บทที่ 2

ทฤษฎีและหลักการทำงานของระบบสกาดาบนระบบปฏิบัติการลีนุกซ์

ภายในเนื้อหาของบทที่ 2 นี้จะกล่าวถึงรายละเอียดของทฤษฎีและหลักการทำงานของระบบสกาดาโดยจะแบ่งเป็นการอธิบายถึงความหมายทั่วไปของระบบสกาดา โครงพื้นฐานของระบบสกาดาตามมาตรฐาน IEEE Standard Definition 37.1 – 1994 โดยสังเขป และการประยุกต์ใช้งานของระบบสกาดา นอกจากนั้นยังได้อธิบายเหตุผล สมมติฐานและข้อกำหนดต่างๆ ของระบบปฏิบัติการลีนุกซ์ที่ใช้งานในระบบสกาดาอย่างเช่นการข้อกำหนดการใช้การสื่อสารแบบมอดบัส (Modbus Protocol) เป็นต้น

2.1 ความหมายของระบบสกาดา (SCADA)

สกาดา (SCADA ย่อมาจากคำว่า Supervisory Control And Data Acquisition) หมายถึง "ระบบ" ที่มีการ

- รวบรวมข้อมูลจากแหล่งต่าง ๆ (Collection of Information)
- ส่งไปที่ศูนย์ควบคุมกลาง (Transferring Data to a Central Site)
- แสดงสถานะการทำงานของอุปกรณ์ในระบบ (Monitor)
- บันทึกข้อมูลที่ได้ในรูปแบบของฐานข้อมูล (Data Acquisition)
- วิเคราะห์และประมวลผล (Analysis for Data Processing) โดยใช้คอมพิวเตอร์
- ส่งผลไปควบคุมการทำงานของอุปกรณ์ต่างๆในระบบได้ (Control)

2.2 โครงสร้างพื้นฐานของระบบสกาดา (SCADA)

ในกรณีทั่วไปอุปกรณ์ควบคุมและวัดบันทึกข้อมูลจะประกอบด้วยสถานีหลัก (Master Terminal Unit หรือ MTU) อย่างน้อย 1 สถานี หน่วยควบคุมปลายทางระยะไกล (Remote Terminal Unit) หรือเรียกอีกชื่อว่า RTU ของสถานีปฏิบัติการ (Field Sites) หนึ่งตัวหรือมากกว่า และมีการเก็บข้อมูลหรือการควบคุมระหว่างสถานีหลักกับหน่วยควบคุมปลายทางระยะไกลโดยการส่งผ่านระบบการสื่อสาร [7] ดังแสดงในรูปที่ 2.1

รูปที่ 2.1 โครงสร้างพื้นฐานของระบบสกาดา

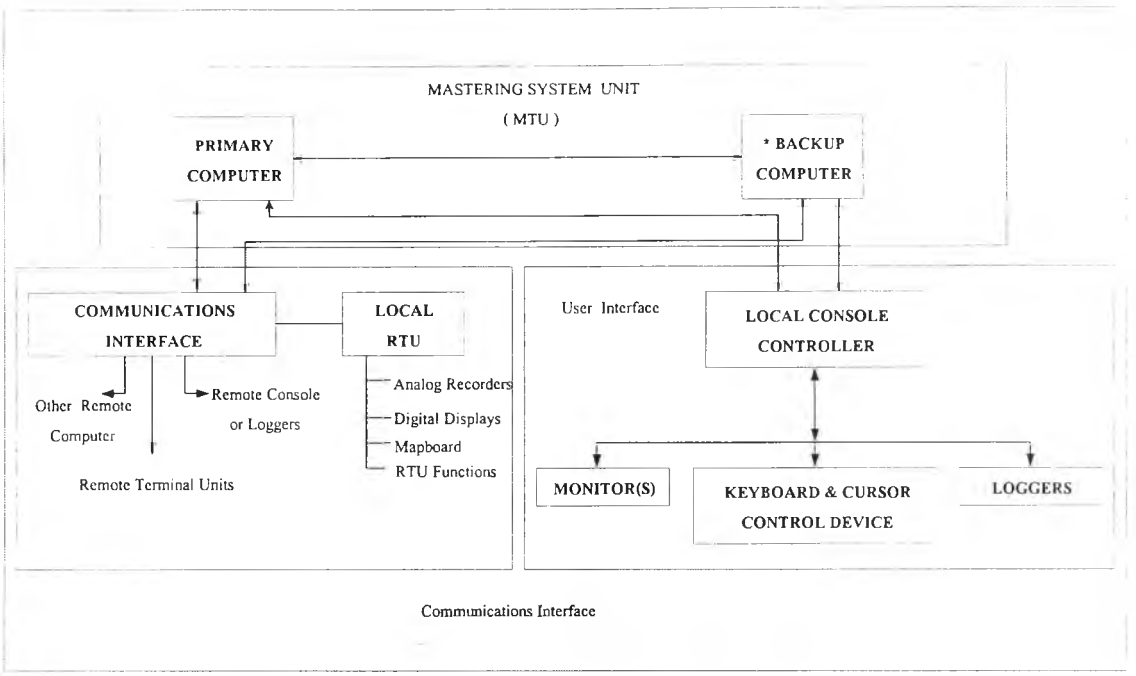
ตามหลักการของระบบสกาดา ([1],[2] อ้างอิงตาม IEEE Standard Definition 37.1 – 1994) จะต้องมีคุณสมบัติโดยสังเขปดังต่อไปนี้

2.2.1 สถานีหลัก (Master Terminal Unit)

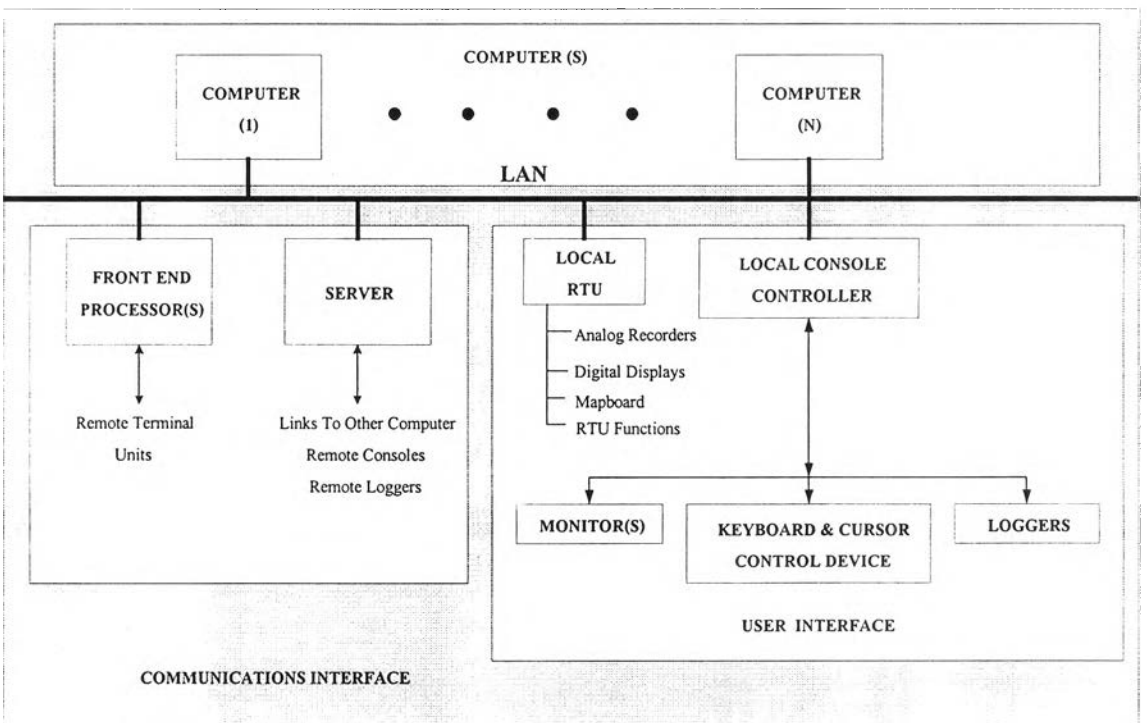
สถานีหลักโดยความหมายแล้วคือจุดรวมของอุปกรณ์ทุกอย่างครบถ้วนตามฟังก์ชันการทำงานและอุปกรณ์อื่นๆที่เชื่อมต่อกันทางไฟฟ้าเพื่อทำงานตามลักษณะฟังก์ชันควบคุมของสถานีควบคุมหลัก อุปกรณ์ทั้งหมดนี้รวมไปถึงจุดเชื่อมต่อกับช่องสัญญาณสื่อสารแต่ไม่รวมช่องสัญญาณที่เชื่อมต่อกันระหว่างการติดต่อสื่อสารกับสถานีควบคุมทางไกลอื่นๆ สถานีหลักจะเป็นจุดที่มีความสำคัญที่สุดในระบบสื่อสาร สามารถแบ่งตามลักษณะการทำงานเป็น 2 แบบ คือ

2.2.1.1 **สถานีหลักแบบรวมฟังก์ชันไว้ที่ศูนย์กลาง** (Centralized System) คือการรวมฟังก์ชันทำงานต่างๆที่ใช้ในระบบสกาดามารวมไว้ที่เครื่องคอมพิวเตอร์เครื่องเดียว (อาจมีการสำรองข้อมูลไว้อีกหนึ่งเครื่องก็ได้)

2.2.1.2 **สถานีหลักแบบกระจายฟังก์ชัน** (Distribution System) คือมีการแบ่งฟังก์ชันการทำงานเป็นกลุ่มๆกระจายออกไปตามหน้าที่การทำงานโดยแบ่งไปอยู่บนคอมพิวเตอร์คนละเครื่อง เช่น เครื่องหนึ่งมีไว้ สำหรับแสดงผลอีกเครื่องหนึ่งมีไว้ ควบคุมและอีกเครื่องเอาไว้เก็บข้อมูลเพื่อการวิเคราะห์เป็นต้น ดังแสดงได้ดังรูปที่ 2.3



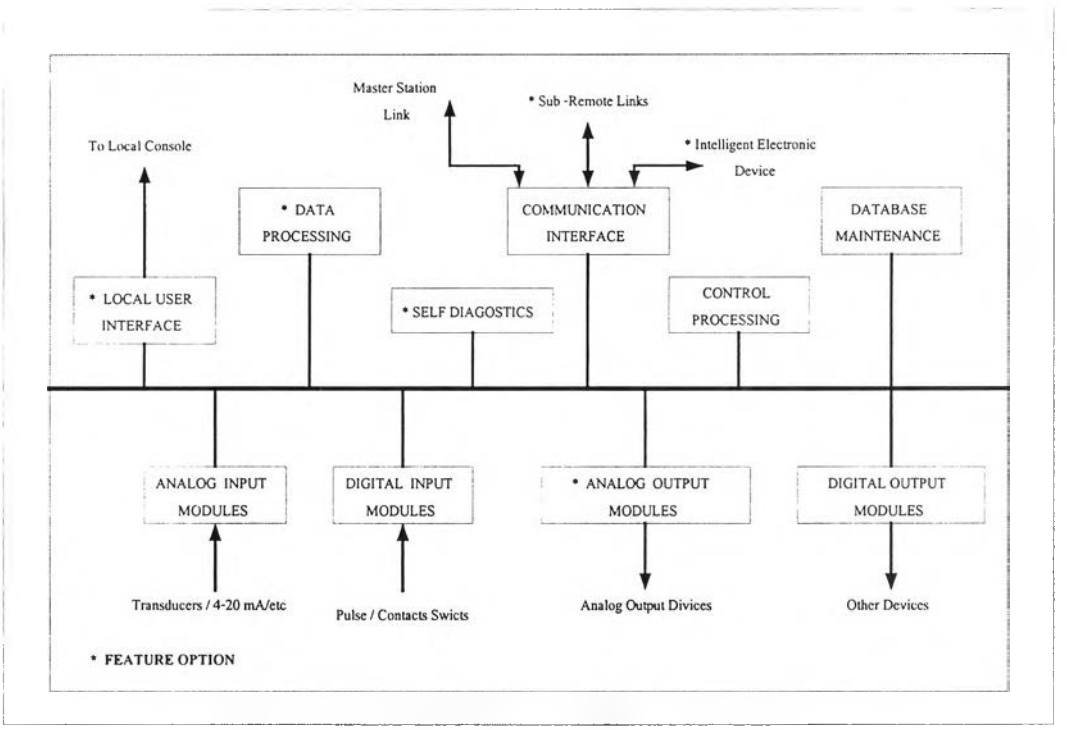
รูปที่ 2.2 การทำงานของสถานีหลักแบบรวมฟังก์ชันไว้ที่ศูนย์กลาง



รูปที่ 2.3 การทำงานของสถานีหลักแบบกระจายฟังก์ชัน

2.2.2 หน่วยควบคุมปลายทางระยะไกล (Remote Terminal Unit)

หน่วยควบคุมปลายทางระยะไกลหรือ RTU โดยความหมายแล้วคือจุดรวมของอุปกรณ์ทุกอย่างครบถ้วน ฟังก์ชันการทำงานและอุปกรณ์อื่นๆที่เชื่อมต่อกันทางไฟฟ้าเพื่อทำงานตามลักษณะฟังก์ชันควบคุมของสถานีควบคุมทางไกล อุปกรณ์ทั้งหมดนี้รวมไปถึงจุดเชื่อมต่อกับช่องสัญญาณสื่อสารแต่ไม่รวมช่องสัญญาณที่เชื่อมต่อกันระหว่างการติดต่อสื่อสารกับสถานีหลัก องค์ประกอบของฟังก์ชันการทำงานของ RTU แสดงได้ดังรูปที่ 2.4



รูปที่ 2.4 ฟังก์ชันการทำงานของ RTU

2.2.3 ระบบสื่อสาร (Communication System)

ระบบการสื่อสารมีหน้าที่หลักในการเชื่อมโยงระหว่างกลุ่มของสถานีหลักกับกลุ่มของ RTU ซึ่งคุณสมบัติโดยสังเขปต้องมีดังต่อไปนี้เป็นอย่างน้อย

- มีการกำหนดการใช้อุปกรณ์ในการรับส่งข้อมูล เช่นโมเด็ม (Modem), วิทยุ (ในย่าน UHF) เป็นต้น
- มีการกำหนดช่องสัญญาณในการรับส่งข้อมูล เช่น สาย LAN
- มีสื่อกลางการสื่อสาร(โพรโตคอล) เช่น MODBUS , ASCII ,RS – 485 เป็นต้น

2.3 การใช้งานของระบบสกาดา

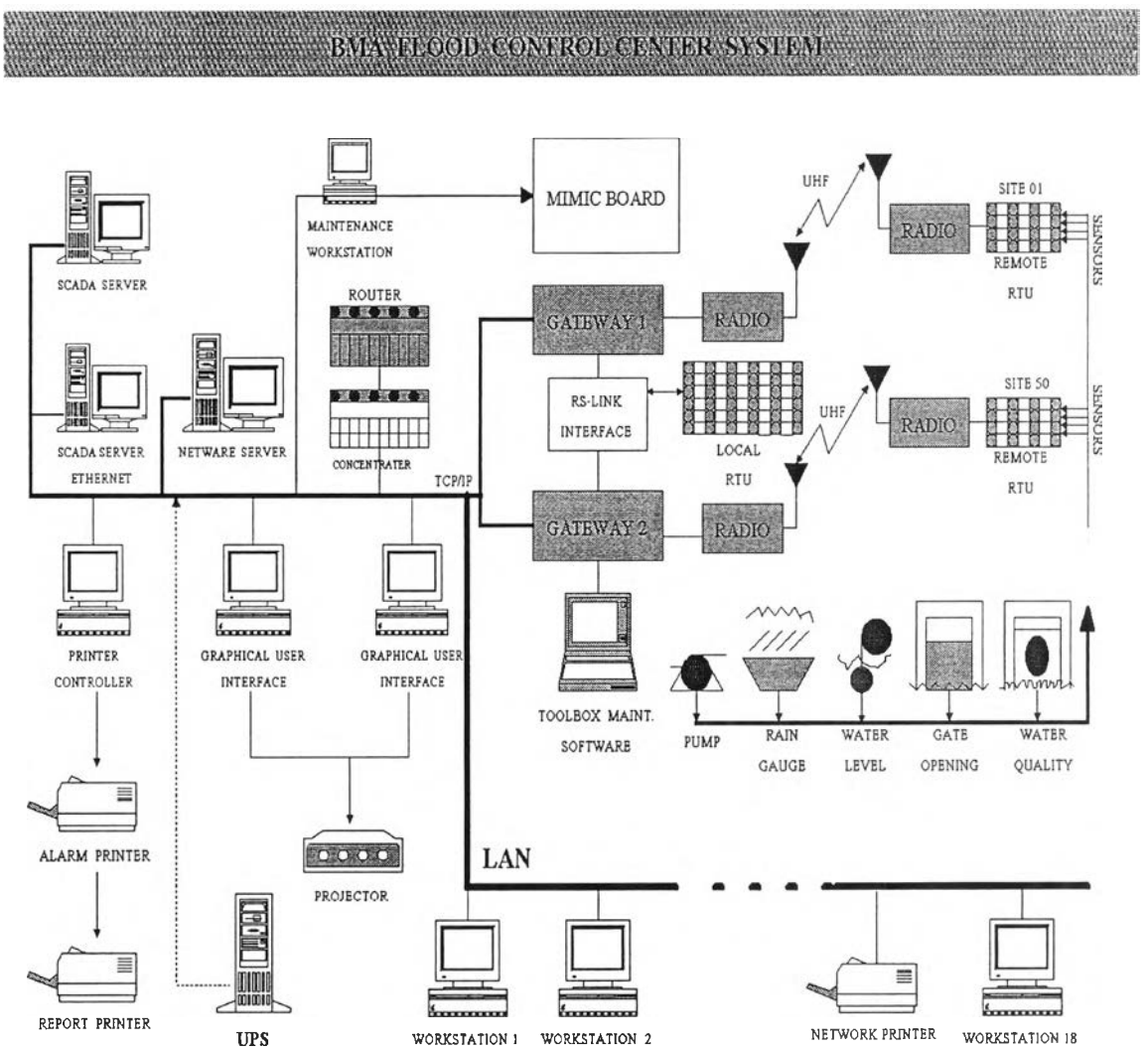
โดยประวัติความเป็นมาของระบบสกาดานั้นวัตถุประสงค์หลักในครั้งแรกจะนำมา ใช้กับงานด้านโทรศัพท์และอุตสาหกรรมการสื่อสารอื่นๆ เป็นระบบที่มีความสามารถในการตรวจสอบการเปลี่ยนแปลงสถานะของอุปกรณ์ที่อยู่ในระยะไกลและรายงานผลการเปลี่ยนแปลงสถานะให้ศูนย์ควบคุมทราบ ต่อมาในปี 1923 ได้มีการพัฒนาระบบการควบคุมระยะไกล โดยการใช้ เทคนิค "Check-before-operate" ทั้งนี้ เพื่อให้มีการตรวจสอบสถานะของอุปกรณ์ที่ต้องการจะควบคุมเพื่อความแน่ใจก่อนที่จะสั่งควบคุมและต่อมาได้มีการนำระบบ "Logging System" มาใช้งาน โดยระบบนี้เป็นระบบที่สามารถเฝ้ามองข้อมูลจากระยะไกลได้และพิมพ์รายงานข้อมูลการเปลี่ยนแปลงสถานะของอุปกรณ์ตลอดจนถึงรายงานวันและเวลาที่เกิดขึ้นด้วย

โดยในปี 1980 ได้มีการนำเทคโนโลยีใหม่เข้ามาใช้ในระบบสกาดาที่ใช้ในการควบคุมระบบไฟฟ้าซึ่งในปัจจุบันนี้ RTU ชนิดใหม่ๆได้มีการนำไมโครโปรเซสเซอร์มาใช้ในการประมวลผลของฟังก์ชันต่างๆ ที่เพิ่มขึ้นมา การนำเอาไมโครโปรเซสเซอร์มาใช้งานนั้นทำให้มีความคล่องตัวในการใช้งานที่เกี่ยวกับระบบสกาดาทั้งในด้านควบคุมการทำงานและความสามารถด้านอื่น ๆในการใช้งาน

ในปัจจุบันระบบสกาดาได้มีการใช้อย่างแพร่หลายทั่วไปในอุตสาหกรรมขนาดกลางโดยการนำระบบควบคุมสกาดามาเป็นระบบควบคุมที่เชื่อมโยงเข้ากับอุปกรณ์ควบคุมในกระบวนการผลิต [6] อาทิเช่น PLC, Multi-loop Controller, Intelligent Transmitter, Digital Power Meter เพื่อให้สามารถแสดงผลการทำงานของสายผลิตในลักษณะกราฟฟิก, แสดงรูปคลื่นค่าสัญญาณในกระบวนการผลิตพร้อมทั้งเก็บประวัติ แจ้งเตือนค่าผิดปกติที่เกิดขึ้นในกระบวนการ ดำเนินการควบคุมตามค่าที่กำหนดล่วงหน้า เพื่อลดข้อผิดพลาดและวัตถุดิบสูญเสียจากพนักงานควบคุม พร้อมทั้งเก็บและพิมพ์รายงานที่เป็นประโยชน์ต่องานควบคุมและคุณภาพของกระบวนการผลิตต่อผู้บริหารโรงงานและในปัจจุบันนี้ยังมีขีดความสามารถให้สิทธิการควบคุมและดำเนินงานระบบผ่านทางเครือข่าย (Network) โดยผ่านทางระบบ LAN หรือ Eternet/Internet ได้อีกด้วย ซึ่งนับว่าเป็นเทคโนโลยีที่อำนวยความสะดวกให้เข้าถึงระบบ สกาดา ต่อผู้บริหารและผู้จัดการโรงงานได้จากทั่วทุกมุมโลกเสมือนอยู่ในห้องควบคุมเช่นเดียวกับวิศวกรทั้งนี้ก็เพื่อให้การจัดการดูแลโรงงานสามารถเกิดข้อมูลสนับสนุนเพื่อการพิจารณาต่อผู้บริหารและจัดการได้ตลอดเวลา

2.3.1 ตัวอย่างระบบสกาดาที่ใช้กันในปัจจุบัน

ตัวอย่างของโครงสร้างระบบสกาดาที่ใช้งานจริงในปัจจุบันของระบบป้องกันน้ำท่วมของกรุงเทพ ฯ [10] ดังต่อไปนี้

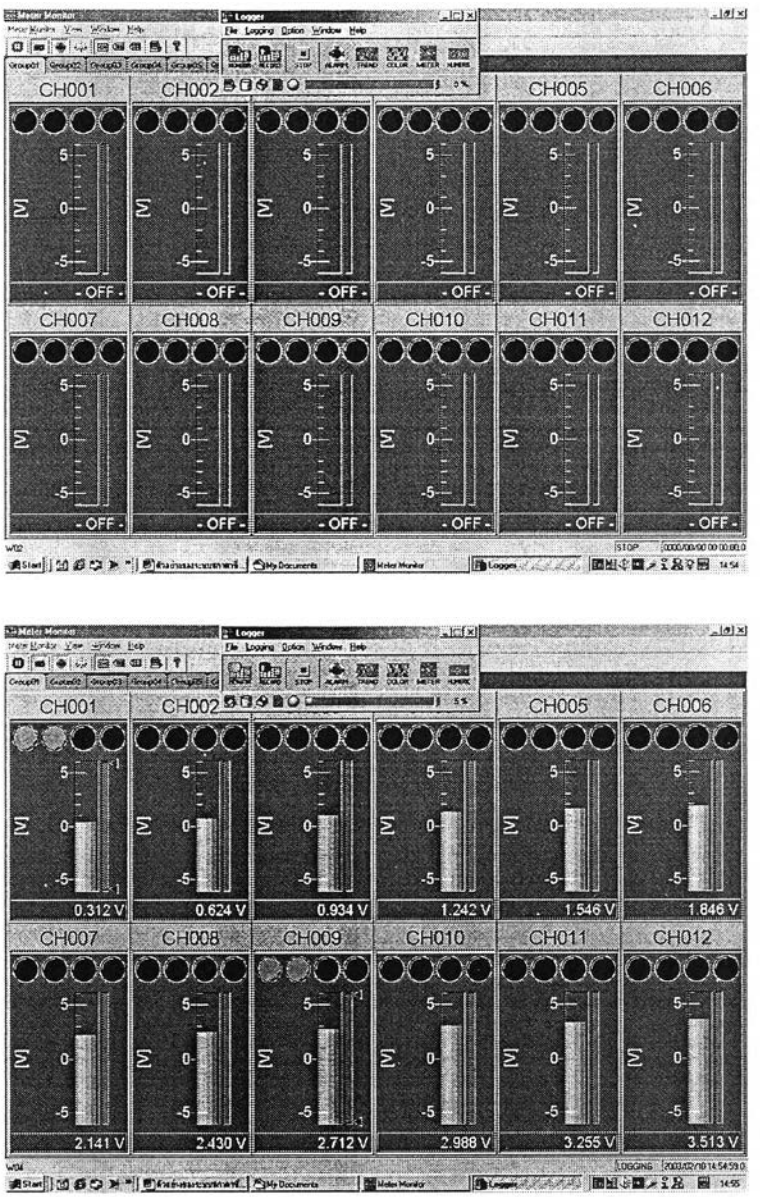


รูปที่ 2.5 โครงสร้างสกาดาของระบบป้องกันน้ำท่วมของกรุงเทพ ฯ [11]

2.3.2 ตัวอย่างของซอฟต์แวร์สกาดา

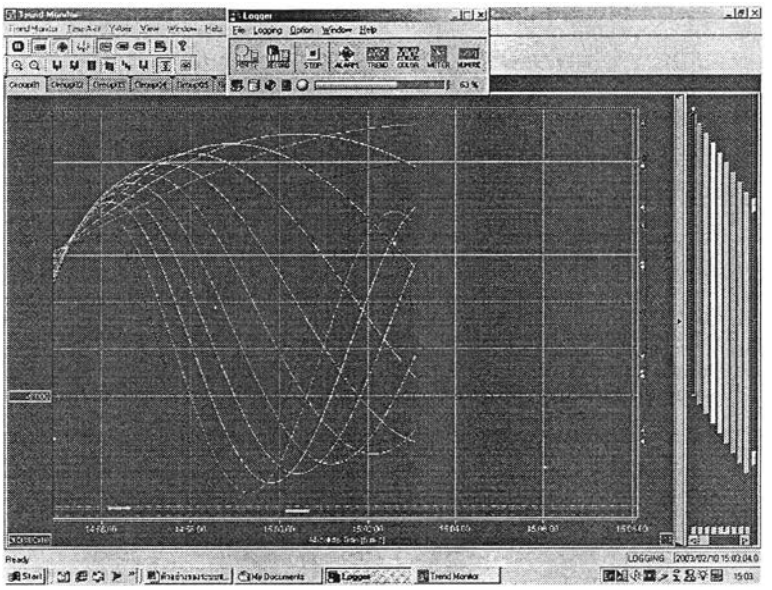
ตัวอย่างของซอฟต์แวร์สกาดาที่ใช้งานจริงของบริษัทโยโกกาวา [12] ซึ่งทำงานบนระบบปฏิบัติการไมโครซอฟต์วินโดว์มีคุณสมบัติในการสื่อความหมายได้อย่างชัดเจนดังนี้

- การวัดค่าที่อ่านเข้ามาด้วยการแสดงในรูปแบบของมาตรวัด (Meter)



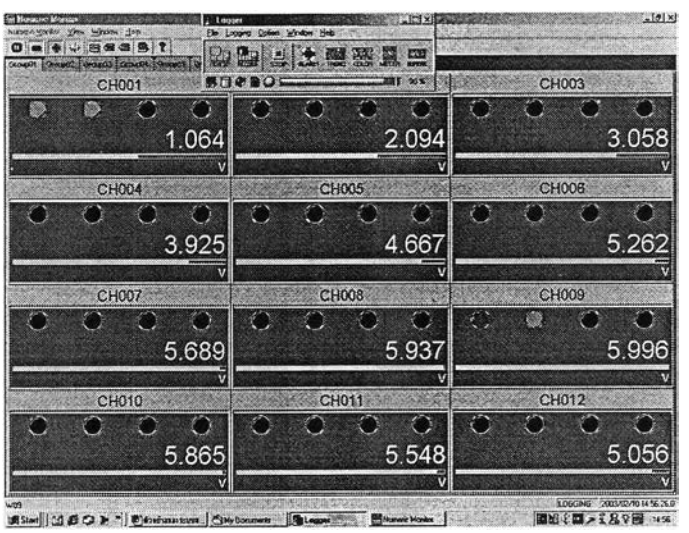
รูปที่ 2.6 การวัดในรูปมาตรวัด

- แสดงผลการวัดค่าที่อ่านเข้ามาด้วยการแสดงในรูปแบบของกราฟ (Trends Graph)



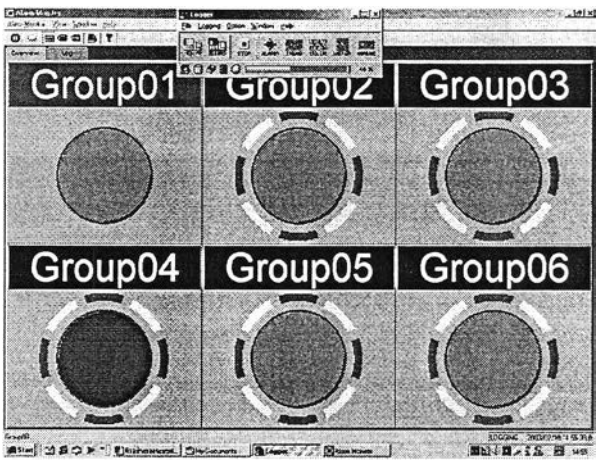
รูปที่ 2.7 การวัดในรูปแบบของกราฟ

- แสดงผลการวัดค่าที่อ่านเข้ามาด้วยการแสดงในรูปแบบของตัวอักษร (Texts) และแสดงถึงสถานะของการการทำงานด้วยหลอดไฟแอลอีดี (LED)



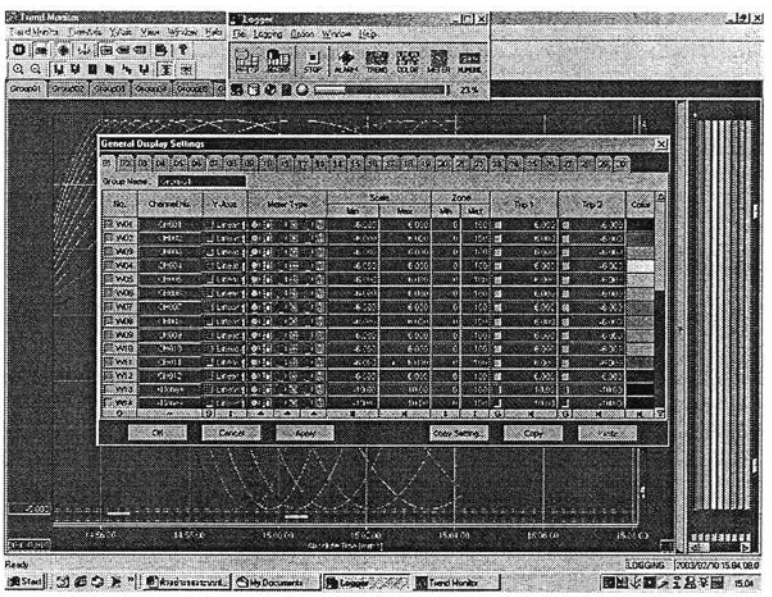
รูปที่ 2.8 แสดงสถานะการทำงานด้วยหลอดไฟแอลอีดีและแสดงผลในรูปตัวอักษร

- แสดงผลการเตือนความผิดปกติด้วยหลอดไฟแอลอีดี (LED)



รูปที่ 2.9 การเตือนความผิดปกติด้วยหลอดไฟแอลอีดี

- แสดงฐานข้อมูลในอดีตในรูปแบบกราฟและไฟล์เอกสาร



รูปที่ 2.10 การดูข้อมูลในอดีต

2.4 ระบบปฏิบัติการลีนุกซ์

2.4.1 พื้นฐานของระบบปฏิบัติการลีนุกซ์

ลีนุกซ์ เป็นระบบปฏิบัติการ (Operating System) ที่ถูกถอดแบบโครงสร้างมาจากระบบปฏิบัติการยูนิกซ์แต่มีส่วนที่แตกต่างกับระบบปฏิบัติการยูนิกซ์ที่ลีนุกซ์ถูกพัฒนาขึ้นมาตามมาตรฐานของ POSIX (Portable Operating System Interface) ซึ่งมีความสามารถในการทำงานกับเครื่องคอมพิวเตอร์ได้หลายตระกูลแต่ยูนิกซ์ส่วนใหญ่จะมีความจำเพาะกับคอมพิวเตอร์บางรุ่นเท่านั้นเช่น SUN SPARC เป็นต้น อีกทั้งลีนุกซ์ยังอยู่ภายใต้ลิขสิทธิ์แบบ GPL (GNU General Public License) ซึ่งหมายถึงการอนุญาตให้นำรหัสต้นฉบับมาทำการแก้ไข ปรับปรุงและแจกจ่ายได้ตามความต้องการอย่างเป็นอิสระ แต่ซอฟต์แวร์นั้นจะต้องยังคงเป็นลิขสิทธิ์แบบ GPL อยู่ [15]

ระบบปฏิบัติการลีนุกซ์ถูกสร้างขึ้นโดยนาย Linus Torvalds ขณะที่ยังเป็นนักศึกษาอยู่ที่มหาวิทยาลัย Helsinki ประเทศฟินแลนด์ โดยเริ่มแรกได้สร้างตามแบบของระบบปฏิบัติการมินิกซ์ (Minix) ก่อน แล้วพัฒนาจนสามารถใช้งานกับเครื่องคอมพิวเตอร์ตระกูล Intel ได้ด้วย จากนั้นได้มีการแจกจ่ายรหัสต้นฉบับกันทางอินเตอร์เนตทำให้มีผู้สนใจกันมากขึ้นและร่วมพัฒนากันอย่างแพร่หลายทั้งในภาคธุรกิจและในมหาวิทยาลัยต่างๆ ในปัจจุบันนี้เราสามารถหาระบบปฏิบัติการลีนุกซ์จากค่ายต่างๆมาใช้ได้สะดวกมากขึ้นเช่น Slackware, Red Hat, Debian, SuSE และ Ziif แต่ยังคงอยู่ภายใต้ GPL เหมือนเดิม

2.4.2 คุณสมบัติของระบบปฏิบัติการลีนุกซ์ที่สนับสนุนการทำงานของระบบสกาดา

1. **ประสิทธิภาพ** ลีนุกซ์มีประสิทธิภาพที่สูงเพียงพอกับความต้องการของระบบสกาดาเนื่องจากลีนุกซ์ได้ถูกออกแบบมาให้ใช้งานอุปกรณ์ฮาร์ดแวร์ทุกอย่างของ เครื่องได้อย่างเต็มประสิทธิภาพ อย่างเช่น
   - **เคอร์เนล** (Kernel) **ของลีนุกซ์สนับสนุนการ** Demand – Paged Loaded Executable **นั่นคือเฉพาะส่วนของโปรแกรมที่กำลังถูกเรียกทำงานเท่านั้นที่จะถูกอ่านจาก ดิสก์เข้าสู่หน่วยความจำของเครื่องทำให้ระบบมีการใช้งานหน่วยความจำอย่างมีประสิทธิภาพ นอกจากนั้นตัวเคอร์เนล ยังสามารถโหลดโปรแกรมขึ้นมาทำงานด้วยวิธี " Shared Copy – on – write pages " คือยอมให้หลายๆ โปรเซสใช้งานในหน่วยความจำเดียวกันได้ซึ่งทำให้ระบบทำงานเร็วขึ้นและลดขนาดการใช้งานในหน่วยความจำได้

- ลีนุกซ์ จะมองอุปกรณ์ฮาร์ดแวร์เป็นไฟล์ดังนั้นการทำงานกับอุปกรณ์จึงมีความคล่องตัวและยืดหยุ่นสูงในการนำไปใช้งาน [14]

2. **เสถียรภาพ** ลีนุกซ์มีการพัฒนาแนวคิดพื้นฐานมาจากระบบ ยูนิกซ์ซึ่งมีชื่อเสียงทางด้านเสถียรภาพ [3]ในการทำงานแต่เพิ่มความสามารถในการทำงานบนคอมพิวเตอร์ได้หลายรุ่นและมีการทำงานแบบระบบหลายผู้ใช้และหลายภารกิจ (Multi – User and Multi – tasking)

3. **การถ่ายโอนข้อมูล** ลีนุกซ์มีความสามารถในการใช้ ไฟล์ร่วมกับระบบปฏิบัติการอื่นๆได้ เช่น FAT32 ของ WIN98 , NTFS ของ WINNT เป็นต้น จึงทำให้สะดวกต่อการถ่ายโอนข้อมูล

4. **ระบบเครือข่าย** ลีนุกซ์สนับสนุนการทำงานกับระบบเครือข่าย (Network System) บน TCP/IP ได้อย่างสมบูรณ์แบบ

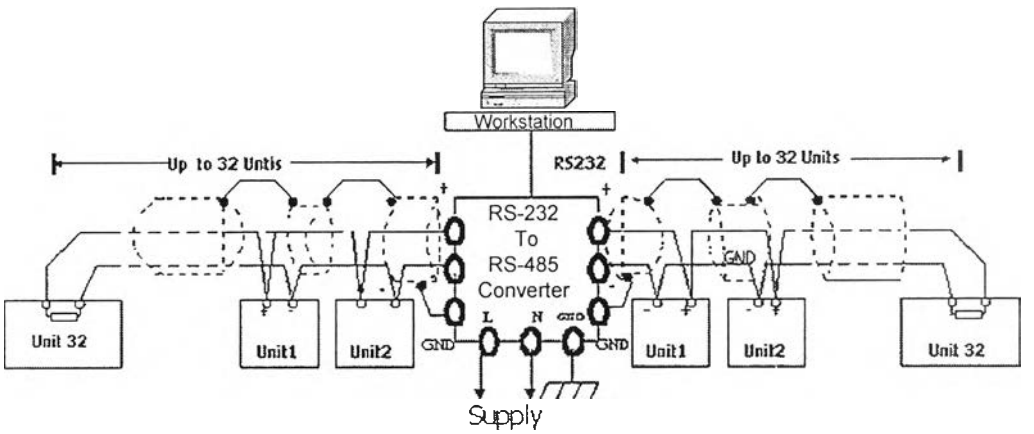
5. **มีผู้ร่วมพัฒนาอยู่ทั่วโลก** มีการแลกเปลี่ยนข้อมูลตลอดเวลาถึงข้อดีและข้อเสียของระบบปฏิบัติการและคำแนะนำในการแก้ปัญหา

6. **ระบบปฏิบัติการแบบเปิด** (Open Source) สิ่งที่สำคัญที่สุดคือลีนุกซ์เป็นระบบปฏิบัติการแบบเปิด (Open Source) ทำให้เราสามารถแก้ไขและเปลี่ยนแปลงทุกอย่างของรหัสต้นฉบับได้ตามความต้องการเหมาะสมกับระบบที่ต้องการพัฒนาให้มีประสิทธิภาพในระยะยาวอย่างมีความต่อเนื่อง

2.5 การสื่อสารทางพอร์ตอนุกรมของระบบสกาดา

วิทยานิพนธ์ฉบับนี้ได้มีข้อกำหนดการสื่อสารข้อมูลระหว่างสถานีหลักกับ RTU เป็นแบบ RS – 485 และใช้โพรโตคอลการสื่อสารเป็นแบบมอดบัสโดยเราเลือกคอมพิวเตอร์ที่ใช้ระบบปฏิบัติการลีนุกซ์ทำหน้าที่เป็นสถานีหลักแบบรวมฟังก์ชันไว้ที่ศูนย์กลาง ดังนั้นการติดต่อระหว่างสถานีหลักกับหน่วยควบคุมปลายทางระยะไกลจะติดต่อกันโดยตรงด้วยมาตรฐาน RS – 485 โดยตรงเลยไม่ได้ ดังนั้นต้องติดตั้งตัวแปลงจากสัญญาณตามมาตรฐาน RS – 232 จาก

พอร์ตอนุกรมของคอมพิวเตอร์เป็นสัญญาณตามมาตรฐาน RS – 485 ก่อนนำไปใช้งานแสดงได้ดังรูปที่ 2.11



รูปที่ 2.11 ลักษณะการทำงานบนมาตรฐาน RS – 485

นอกจากนั้นยังมีรายละเอียดของคุณสมบัติต่างที่ควรทราบดังรายละเอียดต่อไปนี้

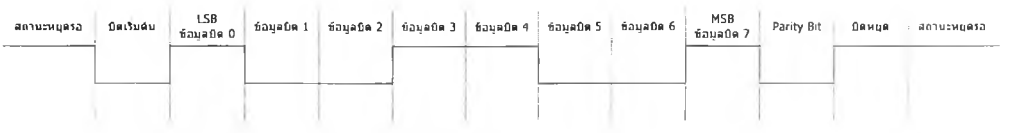
2.5.1 พื้นฐานการสื่อสารข้อมูลอนุกรม

การสื่อสารข้อมูลอนุกรมหมายถึงการรับส่งสัญญาณในลักษณะกลุ่มหรือบิตคราวละหนึ่งบิตเป็นลำดับเรื่อยไปจนสิ้นสุด ซึ่งเหมาะสมกับการสื่อสารในระยะทางไกลเพราะใช้สายส่งสัญญาณน้อยมากโดยเราสามารถแบ่งประเภทการสื่อสารได้ดังต่อไปนี้

2.5.1.1 การสื่อสารแบบไม่ประสานเวลา (Asynchronous Communication)

คือการสื่อสารที่อุปกรณ์ตัวรับกับอุปกรณ์ตัวส่งไม่ได้ใช้เวลาเดียวกันในการส่ง มีการกำหนดของเวลา รับ/ส่ง สัญญาณไม่แน่นอน โดยทางอุปกรณ์ตัวรับจะต้องคอยตรวจสอบสัญญาณเริ่มต้นของอุปกรณ์ตัวส่งทุกครั้ง แม้ว่าการสื่อสารแบบไม่ประสานเวลาจะไม่ต้องการสัญญาณนาฬิกาของฝ่ายส่งแต่ทางฝ่ายรับเองก็ต้องมีสัญญาณนาฬิกาของตนเองเพื่อที่จะใช้ในการพิจารณาแยกอักขระจุดเริ่มต้นและจุดสิ้นสุดของข้อมูลได้

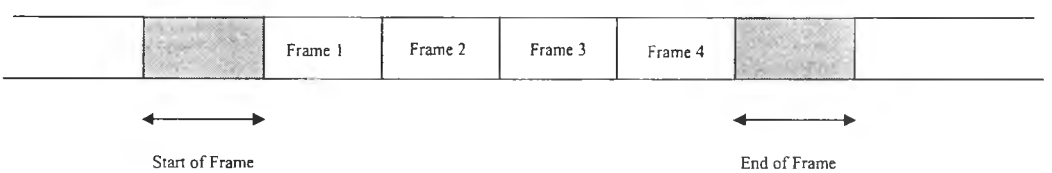
รูปแบบส่วนใหญ่ของการเข้ารหัสแบบไม่ประสานเวลาคือ การเข้ารหัสแบบแอสกี (ASCII Code) โดยมีการส่ง บิตเริ่มต้น (Start bit) 1 บิต ข้อมูล (Data bit) 7 บิต พาริตีบิต 1 บิต และสุดท้ายคือบิตสิ้นสุด (Stop bit) ซึ่งนิยมใช้ 2 บิต แสดงดังรูปที่ 2.12



รูปที่ 2.12 รูปแบบการสื่อสารข้อมูลแบบอะซิงโครนัส

2.5.1.2 การสื่อสารแบบประสานเวลา (Synchronous Communication)

การสื่อสารแบบประสานเวลาคือการส่งสัญญาณระหว่างอุปกรณ์ตัวรับกับอุปกรณ์ตัวส่งที่เวลาเดียวกันหรือที่อุปกรณ์ตัวรับจะต้องมีภาครับที่สามารถสร้างสัญญาณนาฬิกาจากข้อมูลที่ได้รับโดยใช้หลักการของเฟสล็อกลูป (Phase Lock Loop) หรือ ออสซิเลเตอร์ร่วมนัย (Coherent Oscillator) สัญญาณนาฬิกานี้จะใช้เป็นฐานในการรับข้อมูล ดังนั้นการสื่อสารแบบนี้จึงสำคัญที่การเชื่อมต่อสัญญาณนาฬิกาของอุปกรณ์ตัวรับและอุปกรณ์ตัวส่ง ด้วยการสื่อสารแบบนี้จะทำให้ข้อมูลดิจิตอลจะถูกแบ่งเป็นกลุ่มเรียกว่ากรอบ (Frame) แต่ละกรอบจะถูกส่งไปเป็นขบวนโดยไม่มีการหยุดระหว่างแต่ละอักขระและเหมือนกับการสื่อสารแบบไม่ประสานเวลาที่อักขระแต่ละตัวต้องมีบิตเริ่มและบิตสิ้นสุด การสื่อสารแบบประสานเวลาในแต่ละกรอบก็ต้องมีอักขระบอกจุดเริ่มต้นและสิ้นสุดของกรอบนั้นๆและกรอบถัดไป แสดงดังรูปที่2.13



รูปที่ 2.13 รูปแบบการสื่อสารแบบประสานเวลา

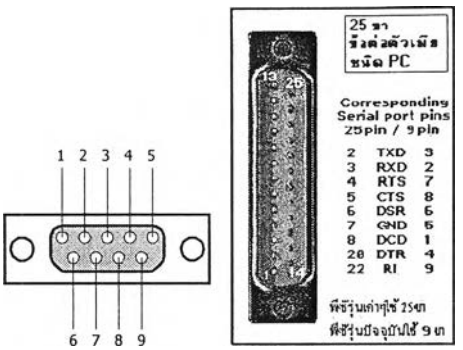
2.5.2 การสื่อสารด้วยมาตรฐาน RS – 232 / RS – 485

2.5.2.1 มาตรฐาน RS - 232

มาตรฐาน RS – 232 เป็นมาตรฐานการสื่อสารผ่านพอร์ตอนุกรมถูกกำหนดโดย EIA (Electronic Industrial Association) ซึ่งเราสามารถแบ่งได้ตามคุณลักษณะได้ 2 ประเภทคือ

คุณสมบัติทางกล

ได้แก่คุณสมบัติของสายไฟและการกำหนดคุณสมบัติจุดข้อต่อรวมถึงระยะทางที่สามารถทำงานได้ดังแสดงในรูปที่ 2.14



รูปที่ 2.14 จุดข้อต่อแบบ DB-9 และ DB-25

ตารางที่ 2.1 รายละเอียดสายสัญญาณของจุดข้อต่อทั้งแบบ DB-9 และ DB-25

คอนเน็กเตอร์ DB-9	คอนเน็กเตอร์ DB-25	ชื่อของสายสัญญาณ	ชนิดของสายสัญญาณ
1	8	Data Carrier Detect : DCD	อินพุต
2	3	Received Data : RxD	อินพุต
3	2	Transmitted Data : TxD	เอาต์พุต
4	20	Data Terminal Ready : DTR	เอาต์พุต
5	7	Signal Ground	-
6	6	Data Set Ready : DSR	อินพุต
7	4	Request to Send : RTS	เอาต์พุต
8	5	Clear to Send : CTS	อินพุต
9	22	Ring Indicator : RI	อินพุต

- Data Carrier Detect : DCD ทำหน้าที่ในการตรวจจับสัญญาณพาหะเพื่อใช้ในการรับข้อมูลที่ส่งเข้ามาที่รีจิสเตอร์บัฟเฟอร์แต่ละครั้ง
- Receive Data : RD **หรือ** RxD ขานี้ใช้เพื่อรับสัญญาณอนุกรมเข้ามายังคอมพิวเตอร์ โดยนำข้อมูลที่อ่านได้เก็บไว้ในรีจิสเตอร์บัฟเฟอร์
- Transmitted Data : TD **หรือ** TxD ขานี้ใช้เพื่อส่งข้อมูลออกจากคอมพิวเตอร์ โดยนำข้อมูลที่เก็บอยู่ในบัฟเฟอร์สำหรับส่งข้อมูลส่งออกไป
- Data Terminal Ready : DTR เป็นขาสัญญาณที่ส่งออกจากคอมพิวเตอร์เพื่อให้อุปกรณ์ปลายทางรับรู้ว่าต้องการติดต่อด้วย โดยขา DTR นี้จะต้องเชื่อมต่อกับขา DSR ของ

อุปกรณ์ปลายทาง และขา DTR ของอุปกรณ์ปลายทางจะต้องเชื่อมต่อกับขา DSR ของคอมพิวเตอร์ ถ้าใช้การเชื่อมต่อเป็นแบบ Null Modem ซึ่งใช้สายในการเชื่อมต่อเพียง 3 เส้น จะต้องต่อขา DTR และ DSR ของตัวมันเองเข้าด้วยกันและต้องต่อกับขา DCD ด้วย ในกรณีที่โปรแกรมสื่อสารที่ใช้มีการตรวจจับสัญญาณพาห์

- Signal Ground : GND ขากราวนด์ของระบบ
- Data Set Ready : DSR ขานี้จะใช้คู่กับขา DTR เพื่อตรวจสอบการเชื่อมต่อกันระหว่างคอมพิวเตอร์กับอุปกรณ์ปลายทาง ซึ่งขา DSR นี้จะเป็นขาสำหรับรับข้อมูลจากภายนอก ซึ่งถูกส่งมาจากขา DTR
- Request to Send : RTS เป็นขาสำหรับส่งสัญญาณร้องขอให้ทางอุปกรณ์ปลายทางส่งข้อมูลกลับมายังคอมพิวเตอร์ โดยขาที่รับสัญญาณ RTS ก็คือขา CTS ในกรณีที่ใช้การเชื่อมต่อแบบ Null Modem 3 สาย จะต้องเชื่อมต่อขา RTS และ CTS ของตัวมันเองเข้าด้วยกัน เพื่อจะให้การรับและส่งข้อมูลสามารถเกิดขึ้นได้ตลอดเวลา
- Clear to Send : CTS ขานี้จะคอยรับสัญญาณจากขา RTS เมื่อรับสัญญาณได้ ข้อมูลที่ขานี้จึงถูกใช้เพื่อตรวจสอบอุปกรณ์ต่อพ่วงว่าพร้อมที่จะรับข้อมูลหรือไม่

คุณสมบัติของสัญญาณไฟฟ้า

1. การสื่อสารในมาตรฐาน RS-232c จะใช้สัญญาณระดับ (Level) ในการสื่อสารข้อมูลโดยอาจพิจารณาในคู่สัญญาณดังต่อไปนี้

MARK / SPACE

OFF / ON

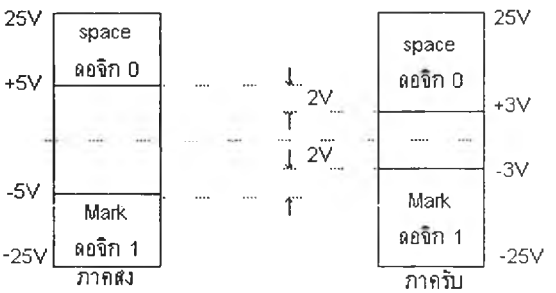
ลอจิก 1 / ลอจิก 0

ระดับของสัญญาณจะแทนด้วยระดับลอจิกลบ (Negative Logic) โดยระดับสัญญาณจะมีลักษณะดังนี้

ตารางที่ 2.2 ลักษณะของระดับสัญญาณ

สถานะ	ระดับแรงดันของสัญญาณ	
	-3V ถึง -25V	3V ถึง 25V
ระดับสัญญาณลอจิก	1	0
	Mark	Space
	Off	On

2. ตัวขับสัญญาณจะต้องส่งสัญญาณระหว่าง -3 ถึง -25V และ 3 ถึง 25 V และยอมให้มีการลดทอนของสัญญาณ ได้ไม่เกิน 2V



รูปที่ 2.15 ระดับสัญญาณของภาครับและภาคส่ง

ในกรณีเช่นนี้แสดงว่าในการรับส่งสัญญาณตามมาตรฐาน RS-232-C จะยอมให้มี noise margin ได้ไม่เกิน 2 V นอกจากนั้น MARK ของ SPACE ยังอาจแทนได้ด้วยการไหลของกระแสไฟฟ้า (Current Loop)

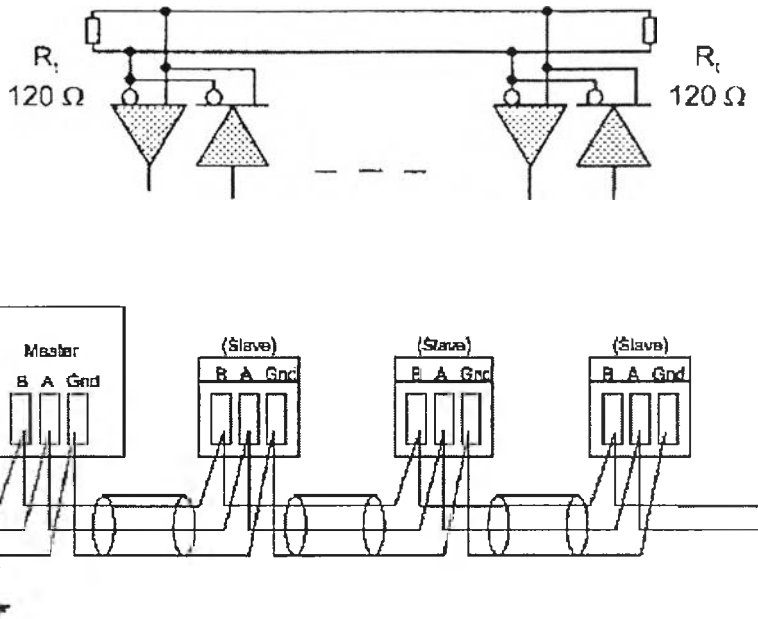
3. ตัวเก็บประจุ CL ซึ่งขนานกับอุปกรณ์ปลายทาง จะต้องมีค่าไม่เกิน 2500 pF โดยไม่รวมค่าความจุของเคเบิล

4. แรงดันไฟฟ้าเมื่อเปิดวงจรจะต้องมีค่าไม่เกิน +/- 25V

5. วงจรรับสัญญาณ RS-232-C จะต้องทนต่อการลัดวงจรของสัญญาณได้โดยไม่ทำให้ภาครับ และอุปกรณ์ที่ต่อพ่วงเสียหาย

2.5.2.2 มาตรฐาน RS – 485

มาตรฐาน RS – 485 เป็นการสื่อสารที่มีอัตราการส่งผ่านข้อมูลสูงสุดที่ 10 Mbps ภายใต้ระยะทาง 4000 ฟุต และสามารถติดต่อกับเครื่องรับ (Receiver) ได้ถึง 32 ตัว โดยระดับของสัญญาณของลอจิก 1 อยู่ที่ –1.5 ถึง –6 V และระดับของสัญญาณของลอจิก 0 อยู่ที่ +1.5 ถึง 6 V ข้อดีของมาตรฐานนี้คือสามารถใช้งานแบบมัลติดรอป (Multidrop Operation) คือสามารถต่อกับเครื่องส่ง (Transmitter) ได้มากกว่าหนึ่งตัวแม้ว่าจะมีเครื่องส่งเพียงเครื่องเดียวที่สามารถทำงานได้ในขณะใดขณะหนึ่ง โดยอาศัยการทำงานของ ไลน์ไดรเวอร์ (Line Driver) ที่ทำหน้าที่คล้ายสภาวะทรี-สเตท (Three-state) คอยสลับการทำงานของเครื่องส่งแต่ละตัว และข้อดีอีกประการหนึ่งคือประหยัดสายในการสื่อสารเพราะใช้เพียง 2 สายต่อกันแบบจุดต่อจุด ดังแสดงในรูปที่ 2.16



รูปที่ 2.16 การเชื่อมต่อกันแบบจุดต่อจุดของมาตรฐาน RS – 485

ตารางที่ 2.3 คุณสมบัติของมาตรฐาน RS – 232 และ RS – 485

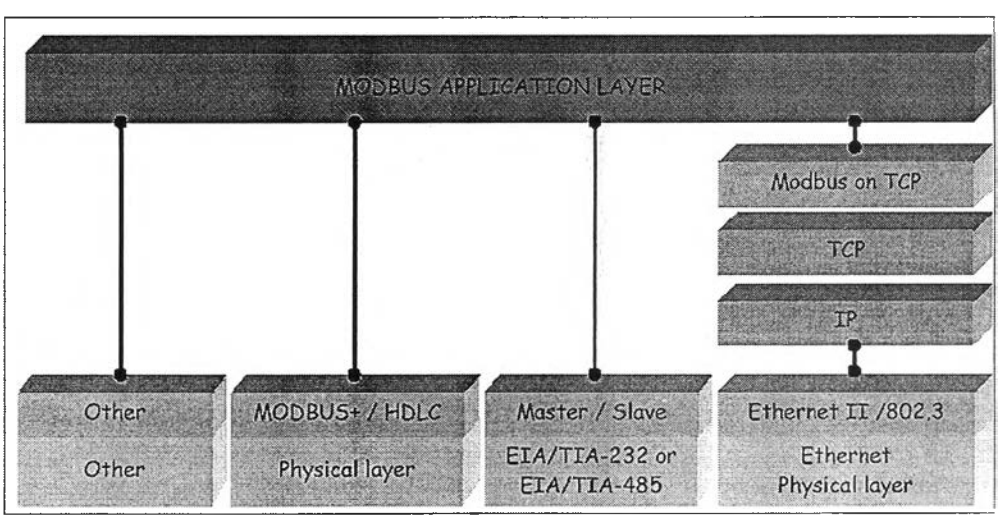
SPECIFICATIONS		RS232	RS485
Mode of Operation		SINGLE -ENDED	DIFFERENTIAL
Total Number of Drivers and Receivers on One Line (One driver active at a time for RS485 networks)		1 DRIVER 1 RECVR	32 DRIVER 32 RECVR
Maximum Cable Length		50 FT.	4000 FT.
Maximum Data Rate (40ft. - 4000ft. for RS422/RS485)		20kb/s	10Mb/s- 100Kb/s
Maximum Driver Output Voltage		+/-25V	-7V to +12V
Driver Output Signal Level (Loaded Min.)	Loaded	+/-5V to +/- 15V	+/-1.5V
Driver Output Signal Level (Unloaded Max)	Unloaded	+/-25V	+/-6V
Driver Load Impedance (Ohms)		3k to 7k	54

ตารางที่ 2.3 (ต่อ) คุณสมบัติของมาตรฐาน RS – 232 และ RS – 485

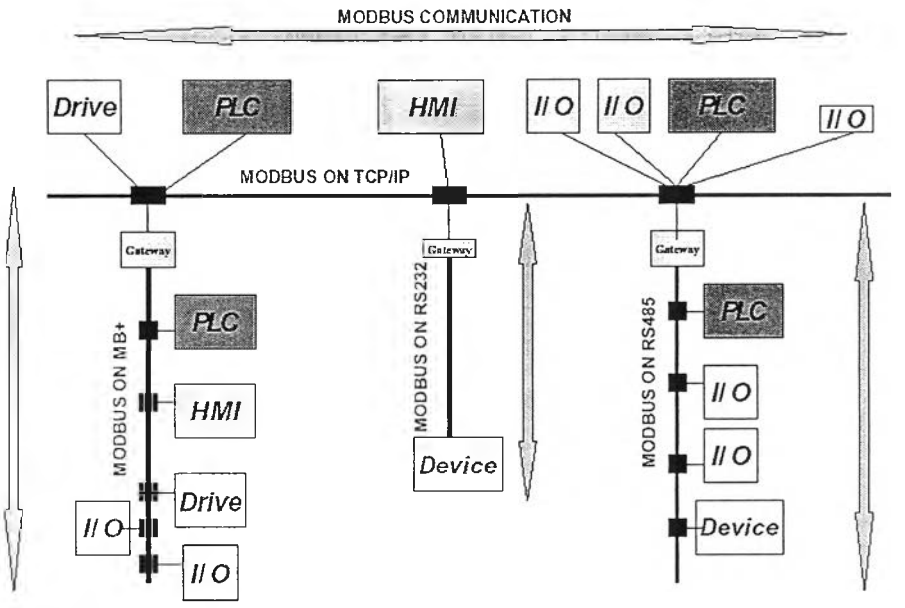
SPECIFICATIONS		RS232	RS485
Max. Driver Current in High Z State	Power On	N/A	+/-100uA
Max. Driver Current in High Z State	Power Off	+/-6mA @ +/-2v	+/-100uA
Slew Rate (Max.)		30V/uS	N/A
Receiver Input Voltage Range		+/-15V	-7V to +12V
Receiver Input Sensitivity		+/-3V	+/-200mV
Receiver Input Resistance (Ohms), (1 Standard Load for RS485)		3k to 7k	>=12k

2.5.3 สื่อกลางการสื่อสารแบบมอดบัส (Modbus Protocol)

มอดบัส [18],[21]เป็นการประยุกต์การใช้งานโพรโตคอลบนชั้นของการส่ง ข้อความ (Layer Messaging Protocol) ในระดับ 7 ของมาตรฐานการเชื่อมต่อระหว่างระบบเปิด (OSI: Open System Interconnection) โดยนำหลักการของไคลเอนท์/เซิร์ฟเวอร์มาใช้เพื่อความคล่องตัวในการที่จะติดต่ออุปกรณ์ต่างชนิดกันบนระบบบัสต่างชนิดกัน



รูปที่ 2.17 การสื่อสารระหว่างอุปกรณ์ต่างชนิดกันบนระบบบัสต่างชนิดกัน

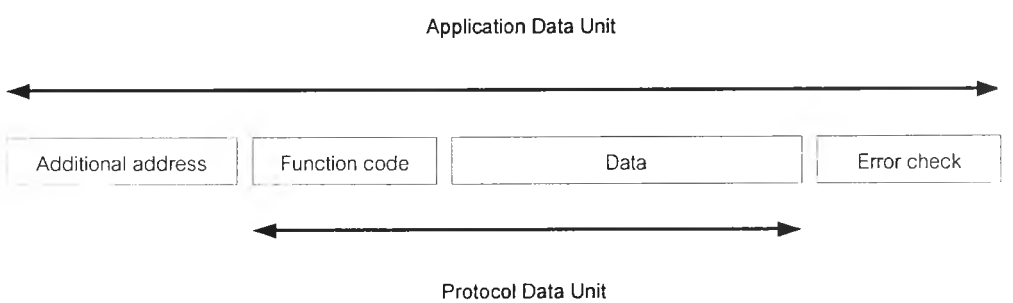


รูปที่ 2.18 การสื่อสารระหว่างอุปกรณ์ต่างชนิดกันบนระบบบัสต่างชนิดกัน

เนื่องจากมอดบัสเป็นสื่อกลางที่มีความซับซ้อนน้อยทำให้ภาคอุตสาหกรรมนำไปใช้กันอย่างแพร่หลายตั้งแต่ ปี ค.ศ. 1979 เป็นต้นมา จนบัดนี้มีอุปกรณ์ที่สนับสนุนการใช้งานโพรโตคอลนี้อยู่ทั่วโลกนับล้านชิ้น

รายละเอียดของมอดบัสโพรโตคอล

1. โครงสร้างพื้นฐานของการส่งข้อมูลแบบมอดบัส



รูปที่ 2.19 โครงสร้างพื้นฐานของการส่งข้อมูลแบบมอดบัส

โดยโครงสร้างพื้นฐาน จะสามารถแบ่งส่วนสำคัญได้สองส่วนคือ

1.1 หน่วยบรรจุข้อมูลของโพรโตคอล (Protocol Data Unit) มีหน้าที่ในการบรรจุส่วนสำคัญสองส่วนคือ รหัสการทำงาน (Function code) และข้อมูล (Data) ที่จะให้สถานีปลายทางปฏิบัติตามหน่วยบรรจุข้อมูลของโพรโตคอลนี้จะมีรูปแบบค่อนข้างตายตัวไม่ว่าการประยุกต์การใช้บนมาตรฐานใดเช่น TCP/IP หรือ RS-485 เป็นต้น

1.2 หน่วยการประยุกต์ข้อมูล (Application Data Unit) ในส่วนนี้จะประกอบด้วยส่วนแรกเป็นโครงสร้างหลักแล้วใส่ข้อมูลเลขหมายของสถานีปลายทางที่จะติดต่อ (Additional address) รวมถึงรหัสสำหรับการตรวจสอบความผิดพลาดจากการส่งข้อมูล (Error check) หน่วยการประยุกต์ข้อมูลนี้จะมีโครงสร้างไม่แน่นอนโดยจะขึ้นอยู่กับมาตรฐานที่ใช้ในการสื่อสารแบบใด

2. รูปแบบการทำงาน

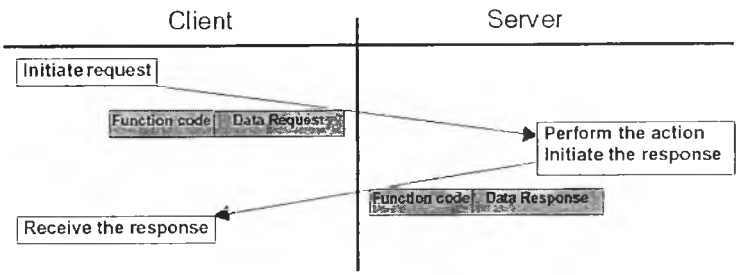
อย่างที่กล่าวไปแล้วในตอนต้นว่ามอดบัสโพรโตคอลอาศัยหลักการทำงานของระบบไคลเอนท์/เซิร์ฟเวอร์ในการทำงานหรือถ้าจะมองตามลักษณะการใช้งานก็จะสามารถเปรียบเทียบได้ดังนี้

เซิร์ฟเวอร์ (Server) $\longmapsto$ สถานีปฏิบัติการ เช่น PLC เป็นต้น

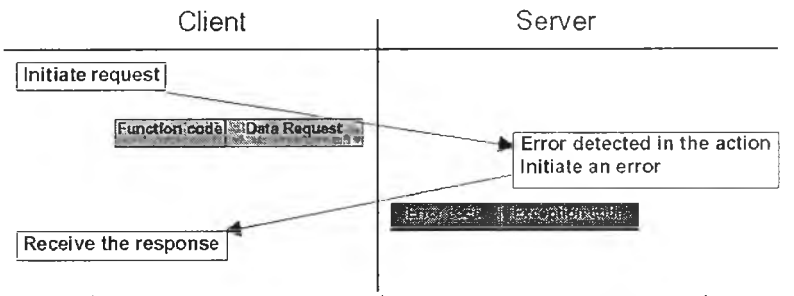
ไคลเอนท์ (Client) $\longmapsto$ ส่วนควบคุมและแสดงผลเช่น คอมพิวเตอร์ เป็นต้น

แต่ก็มีข้อยกเว้นว่าบางครั้งไคลเอนท์ก็สามารถทำงานเป็นระบบเซิร์ฟเวอร์ ได้ถ้าตัวมันเองแสดงพฤติกรรมเป็นผู้ถูกร้องขอการใช้งาน และ บางครั้งระบบเซิร์ฟเวอร์ก็สามารถทำงานเป็นไคลเอนท์ได้ถ้าตัวมันเองแสดงพฤติกรรมเป็นผู้ร้องขอการใช้งาน

ในขณะเริ่มทำงานนั้นจะเริ่มครั้งแรกที่ไคลเอนท์ก่อน โดยการส่งสัญญาณร้องขอให้ตอบสนองต่อรหัสการทำงานและข้อมูลที่ส่งไปจากนั้นเซิร์ฟเวอร์จะทำการตรวจสอบรหัสการทำงานและข้อมูลที่ส่งมาว่าถูกต้องหรือไม่ถ้าถูกต้องจะปฏิบัติตามและส่งข่าวสารไปว่าได้ปฏิบัติการเสร็จสิ้นแล้ว ดังแสดงตามรูปที่ 2.20 แต่ถ้าเกิดการผิดพลาดขึ้นมาเซิร์ฟเวอร์จะส่งรหัสความผิดพลาดกลับไปบอกถึงสาเหตุความผิดพลาด ดังแสดงตามรูปที่ 2.21



รูปที่ 2.20 การทำงานในสภาวะปกติ



รูปที่ 2.21 การทำงานในขณะที่เกิดความผิดพลาดของชุดคำสั่ง

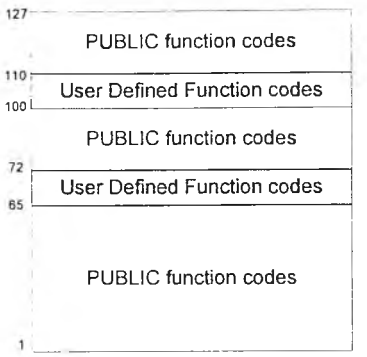
3. ข้อกำหนดมาตรฐานของรหัสการทำงาน (Function code)

ตามมาตรฐานการทำงานมอดบัสมีข้อกำหนดรหัสการทำงานดังต่อไปนี้

- รหัสการทำงานแบบสาธารณะ (Public function codes) คือรหัสที่เป็นมาตรฐานสากลที่มีผู้ใช้ร่วมกันโดยทั่วไปโดยถ้าเปรียบเทียบให้อยู่บนเลขฐานสิบแล้วรหัสควบคุมการทำงานชนิดนี้จะมีค่าได้อยู่ในช่วง 1- 64, 73 - 99 และ 111 – 127 ดังรูปที่ 2.22

- รหัสการทำงานที่ผู้ใช้กำหนดขึ้นเอง (User Defined Function codes) เป็นรหัสที่ผู้ใช้งานกำหนดขึ้นมาเองตามความเหมาะสมที่นำไปใช้งาน มีรูปแบบที่ไม่เป็นมาตรฐานโดยผู้ใช้สามารถกำหนดได้ในช่วงต่อไปนี้ 65 – 72 และ 100 - 110

- รหัสการทำงานที่บริษัทผู้ผลิตกำหนดขึ้นเอง (Reserved function codes) เป็นรหัสที่มีใช้กับผลิตภัณฑ์ของบริษัทผู้ผลิตเท่านั้นไม่เป็นมาตรฐานโดยทั่วไปและไม่อยู่ในช่วง 1 – 127 ซึ่งอาจเป็นตัวเลขหรือตัวอักษรก็ได้



รูปที่ 2.22 ข้อกำหนดของรหัสการทำงานแบบสาธารณะและผู้ใช้กำหนดเอง

ตารางที่ 2.4 ข้อกำหนดของรหัสการทำงานแบบสาธารณะ

				Function Codes		
				code	*Sub code*	*(hex)*
Data Access	Bit access	Physical Discrete Inputs	Read Input Discrete	02		02
		Internal Bits Or Physical coils	Read Coils	01		01
			Write Single Coil	05		05
			Write Multiple Coils	15		0F
	16 bits access	Physical Input Registers	Read Input Register	04		04
		Internal Registers Or Physical Output Registers	Read Multiple Registers	03		03
			Write Single Register	06		06
			Write Multiple Registers	16		10
			Read/Write Multiple Registers	23		17
			Mask Write Register	22		16
	File record access		Read File record	20	6	14
			Write File record	21	6	15
Encapsulated Interface			Read Device Identification	43	14	2B

ในวิทยานิพนธ์ฉบับนี้เลือกใช้การประยุกต์ของมอดบัสบนมาตรฐาน RS-485 แบบแอสกี (Modbus ASCII On RS-485) ซึ่งมีรูปแบบการใช้งานดังนี้

ADDR	FUNCTION	DATA	ERROR CHECK	EOF	READY TO REC RESP
2- CHAR 16 - BITS	2- CHAR 16 - BITS	N x 4- CHAR N x 16 - BITS	2- CHAR 16 - BITS	CR	LF

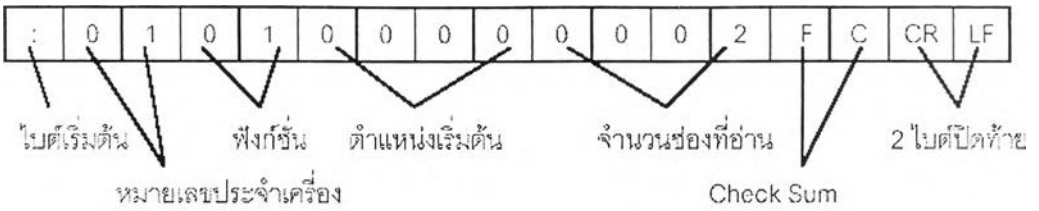
รูปที่ 2.23 รูปแบบโครงสร้างในการส่งข้อมูลไปยัง RTU

ในการส่งข้อมูลแต่ละครั้งจะประกอบไปด้วย

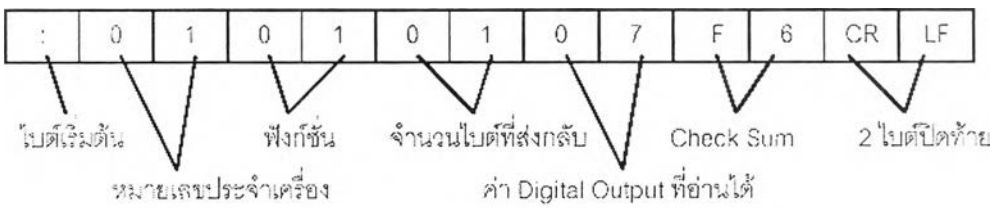
- หมายเลขของอุปกรณ์ที่จะติดต่อจำนวน 2 ไบต์
- รหัสการทำงานจำนวน 2 ไบต์
- ข้อมูล (ขึ้นกับชนิดของอุปกรณ์ว่าจะมีความยาวเท่าไร)
- การตรวจสอบความผิดพลาดแบบการตรวจด้วยส่วนซ้ำซ้อนตามยาว (Longitudinal Redundancy Check(LRC)) จำนวน 2 ไบต์ หรือบางอุปกรณ์อาจใช้การตรวจสอบด้วยส่วนซ้ำซ้อนแบบวน (Cyclic Redundancy Check (CRC)) ก็ได้
- สุดท้ายคือจุดสิ้นสุดของเฟรม (End Of Frame (EOF))

นอกจากนั้นชนิดข้อมูลที่ส่งแต่ละครั้งของชุดข้อมูลก็จะประกอบด้วย 1 บิต สำหรับบิตเริ่ม (Start bit) 7 บิตสำหรับบิตข้อมูล (Data bit) 1 บิตสำหรับพาริตีบิต และอีก 1 บิต สำหรับบิตหยุด (Stop bit) แต่อาจเป็น 2 บิตถ้าไม่มีการใช้พาริตีบิต ดังแสดงได้ตามรูปที่ 2.24

รูปที่ 2.24 รูปแบบของการส่งข้อมูลแต่ละครั้ง



รูปที่ 2.25 ตัวอย่างการส่งชุดคำสั่งไปยัง RTU [22]



รูปที่ 2.26 ตัวอย่างการตอบกลับจาก RTU

รูปที่ 2.25 แสดงตัวอย่างการขอให้สถานีปฏิบัติการที่ 1 อ่านค่า ดิจิตอลเอาต์พุตที่ 1-3 และรูปที่ 2.26 แสดงถึงผลจากการปฏิบัติตามคำขอเสร็จเรียบร้อยแล้ว

สรุปท้ายบท

โดยเนื้อหาภายในบทที่ 2 ทำให้ทราบถึงรายละเอียดของข้อกำหนดและทฤษฎีของระบบสกาดาและคุณสมบัติของระบบปฏิบัติการลีนุกซ์รวมถึงการใช้งานตามมาตรฐาน RS-485 ด้วยโพรโตคอลมอดบัส ในบทต่อไปจะเป็นการอธิบายถึงการออกแบบระบบสกาดาบนระบบปฏิบัติการลีนุกซ์โดยจะกล่าวถึงรายละเอียดของส่วนสำคัญสำหรับการทำงานเป็นระบบสกาดาที่มีสถานีหลักเป็นแบบรวมฟังก์ชันไว้ที่ศูนย์กลางและมีการติดต่อกับสถานีปฏิบัติการซึ่งเป็นแหล่งรวม RTU โดยสื่อสารกันด้วยมาตรฐาน RS-485 อย่างเช่นการใช้พอร์ตสื่อสารอนุกรมสื่อสารกันด้วยโพรโตคอลมอดบัสจะต้องทำการกำหนดอะไรบ้างรวมถึงลักษณะการใช้ฐานข้อมูลสำหรับแสดงผลเป็นแบบใด