บทที่ 3

การออกแบบโครงสร้างระบบสกาดาบนระบบปฏิบัติการลีนุกซ์

ในบทที่ 2 ได้กล่าวถึงทฤษฎีพื้นฐานของระบบสกาดาและระบบปฏิบัติการลีนุกซ์ซึ่งเป็นส่วนสำคัญที่ใช้ในการออกแบบโครงสร้างการใช้งานของระบบสกาดา ส่วนบทนี้จะเป็นการอธิบายถึงการออกแบบโครงสร้างระบบสกาดาบนระบบปฏิบัติการลีนุกซ์ที่มีคุณลักษณะของสถานีหลักเป็นแบบรวมฟังก์ชันไว้ที่ศูนย์กลางและมีการติดต่อกับสถานีปฏิบัติการซึ่งเป็นแหล่งรวม RTU โดยสื่อสารกันด้วยมาตรฐาน RS-485 โดยรายละเอียดภายในบทนี้เป็นการกำหนดคุณสมบัติต่างๆให้กับระบบสกาดาโดยทำการกำหนดโครงสร้างแบบจำลองที่ต้องการขึ้นมาก่อนต่อจากนั้นจะทำการอธิบายรายละเอียดของการสร้างระบบจำลองทีละขั้นตอนรวมถึงข้อกำหนดต่างๆในการใช้ทรัพยากรของระบบปฏิบัติการลีนุกซ์

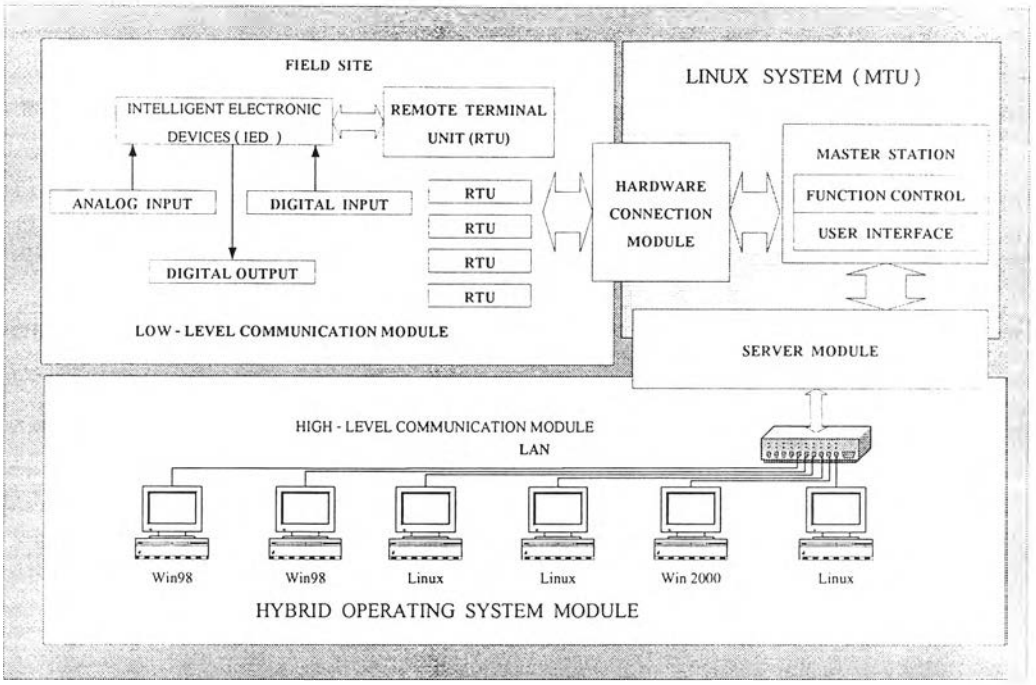
3.1 โครงสร้างแบบจำลองของระบบสกาดาบนระบบปฏิบัติการลีนุกซ์

การออกแบบระบบสกาดาบนระบบปฏิบัติการลีนุกซ์ที่นำมาทดสอบการทำงานถูกออกแบบให้เป็นระบบสกาดาแบบสถานีหลักเป็นแบบรวมฟังก์ชันไว้ที่ศูนย์กลางและมีการติดต่อกับสถานีปฏิบัติการซึ่งเป็นแหล่งรวม RTU โดยสื่อสารกันด้วยมาตรฐาน RS-485 โดยการโพลลิ่งด้วยโพรโตคอลสื่อสารแบบ MODBUS (ASCII) และสามารถตรวจสอบสถานะการทำงาน (Monitoring) ได้โดยผ่านระบบปฏิบัติการไมโครซอฟต์วินโดว์และลีนุกซ์ ดังแสดงไว้ตามรูปที่ 3.1 และมีรายละเอียดการออกแบบดังต่อไปนี้

3.1.1 โครงสร้างของสถานีหลัก

สถานีหลักในวิทยานิพนธ์ฉบับนี้ถูกจำลองขึ้นบนระบบปฏิบัติการลีนุกซ์ซึ่งโดยสมมติฐานจะต้องประกอบด้วยส่วนการทำงาน ดังต่อไปนี้

1. ส่วนต่อประสานกับผู้ใช้ (User Interface) ทำหน้าที่ในการตอบสนองความต้องการกับผู้ใช้ตามคุณสมบัติของระบบสกาดา
2. ฟังก์ชันควบคุม(Function Control) ทำหน้าที่ในการควบคุมการทำงานของ RTU เพื่อที่จะเก็บข้อมูลเข้าฐานข้อมูลแสดงผลในรูปกราฟฟิกหรือตัวอักษรและประสานงานกับส่วนต่อประสานกับผู้ใช้



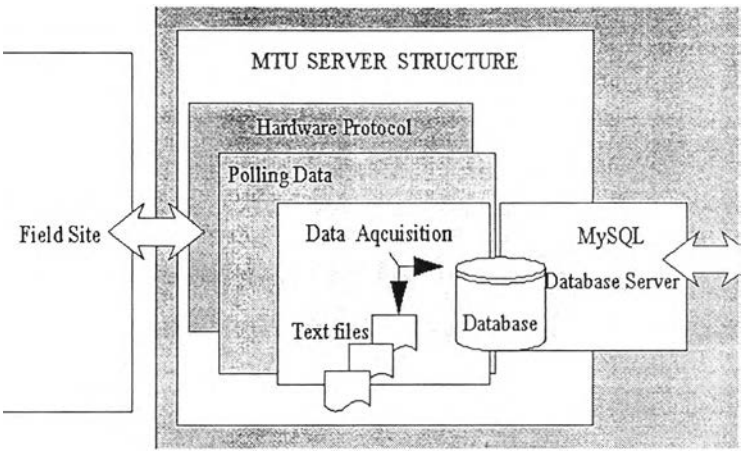
รูปที่ 3.1 โครงสร้างที่ใช้ทดสอบการทำงานของลีนุกซ์สกาดา

3. หน่วยเชื่อมโยงฮาร์ดแวร์(Hardware Connection Module) ทำหน้าที่ในการควบคุมการรับส่งข้อมูลและแปลงข้อมูลจากข้อมูลดิบเป็นข้อมูลที่สามารถนำไปใช้ในรูปแบบต่างๆได้
4. หน่วยบริการเครือข่าย (Server Module) ทำหน้าที่ในการนำข้อมูลจากฐานข้อมูลในสถานีหลักออกมาเผยแพร่ให้หน่วยระบบปฏิบัติการแบบผสมผสาน (Hybrid Operating System Module) นำไปใช้ได้

จากข้อกำหนดและคุณสมบัติที่กล่าวมาสามารถนำมาแจกแจงและออกแบบเพื่อปฏิบัติตามจุดประสงค์การทำงานของสถานีหลักบนระบบปฏิบัติการลีนุกซ์จนสามารถจำลองเป็นโครงสร้างของสถานีหลักที่สามารถทำงานได้ตามจุดประสงค์ดังแสดงในรูปที่ 3.2 และมีรายละเอียดของส่วนประกอบการทำงานดังต่อไปนี้

ฮาร์ดแวร์โพรโตคอล (Hardware Protocol)

เป็นส่วนของโปรแกรมที่ใช้บรรจุโพรโตคอลการสื่อสารข้อมูลที่ถูกเรียกใช้โดยการโพลลิ่งข้อมูลโดยการทดสอบนี้เลือกใช้มอดบัสแอสกี (Modbus ASCII) บนมาตรฐานการส่งสัญญาณแบบ RS-485 และสามารถเพิ่มการเรียกใช้มาตรฐานการส่งสัญญาณแบบอื่นได้ในส่วนนี้



รูปที่ 3.2 โครงสร้างการทำงานของสถานีหลัก

การโพลลิ่งข้อมูล (Polling Data)

ในส่วนนี้จะทำหน้าที่ในการดึงข้อมูลจาก RTU โดยใช้โพรโตคอลที่กำหนดไว้ในส่วนของฮาร์ดแวร์โพรโตคอล รวมทั้งทำหน้าที่ในการตรวจสอบสถานภาพการติดต่อระหว่าง RTU กับสถานีหลักเพื่อค้นหาความผิดปกติที่อาจเกิดขึ้นได้ เช่น สายหลุด เป็นต้น และสามารถตรวจสอบความถูกต้องของข้อมูลที่รับเข้ามาให้เป็นไปตามรูปแบบที่กำหนด ก่อนที่จะนำข้อมูลที่ได้บันทึกลงในฐานข้อมูลต่อไป

การเก็บบันทึกข้อมูล (Data Acquisition)

เป็นการนำข้อมูลที่ผ่านการตรวจสอบแล้วมาบันทึกลงในฐานข้อมูล 2 ประเภท คือฐานข้อมูลที่ทำหน้าที่ช่วยในการแสดงผล และฐานข้อมูลที่ใช้สำหรับบันทึกข้อมูลอย่างเดียว ซึ่งจะกล่าวถึงในส่วนของการทำงานกับฐานข้อมูลต่อไป ในส่วนนี้นอกจากจะทำหน้าที่ในการเก็บบันทึกข้อมูลแล้วยังสามารถนำข้อมูลที่เก็บไว้กระจายให้มีการใช้ข้อมูลร่วมกันในรูปแบบของฐานข้อมูลเซิร์ฟเวอร์ซึ่งสนับสนุนการทำงานแบบหลายผู้ใช้ (Multi-User) ได้อีกด้วย

3.1.2 โครงสร้างของ RTU

RTU ทำหน้าที่ในการติดต่อควบคุมกับการติดต่อสื่อสารอุปกรณ์ปลายทางฉลาด (Intelligent Electronic Devices) ซึ่งรวมถึงอุปกรณ์ที่รับส่งสัญญาณแอนะล็อกและดิจิตอลด้วย เพื่อรับส่งข้อมูลให้กับสถานีหลักในการทดสอบนี้ได้เลือกอุปกรณ์ที่มีผู้พัฒนาแล้วและมีจำหน่ายอยู่มาเป็น RTU โดยเลือก AI210 ของบริษัท Wisco ซึ่งเป็น Analog Module ที่มีคุณสมบัติดังต่อไปนี้

FEATURES

- 8 Analog Input (Progammable Analog Input) 13 1 - (Isolation) Analog Relay, Digital OPTOELECTRONICS RS232/RS485 (RS232 to RS485 Converter)
- Communicate Protocol
  - MODBUS (ASCII) Protocol Compatible

SPECIFICATIONS

- Acquisition Time : 0.3 (300ms)

Digital Input

- Input Range : 0 – 24 V (Low = 0 – 3 V, High = 4 V)
- Impedance : 1000 (1 K)

Analog Input

- Accuracy : 1
- Thermocouple : R, S, K, E, J, T, B
- R.T.D. : Pt100 3-wire type
- Voltage : 0-10V, 0-5V, 0-100mV
- Current : 0-20mA, 0-40mA ()

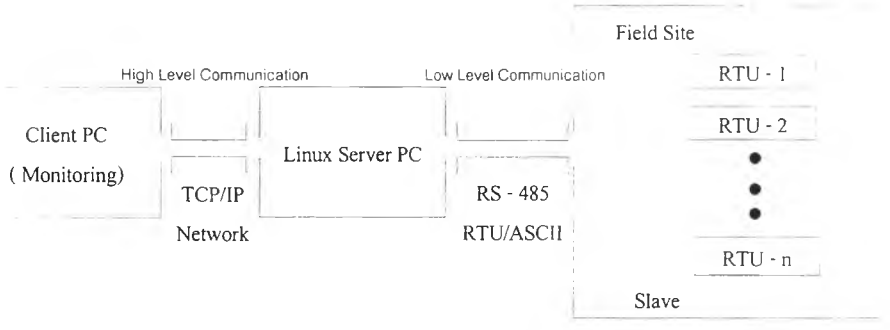
Digital Output

- Output Type : Open Collector Transistor 50V, 500mA

Communication

- Signal Level : EIA Standards, RS-232 and RS-485
- Data Format : 8 data bit, 1 stop bit, none parity
- NO. Station : 0 - 255 (00H - FFH)
- Baud Rate : 1200, 2400, 4800, 9600, 14400, 19200, 28800, 38400 and 57600 bps.
- Distance : RS-232 20m, RS-485 500m

3.1.3 โครงสร้างของระบบสื่อสาร (Communication System)



รูปที่ 3.3 โครงสร้างของระบบสื่อสาร

โครงสร้างของระบบสื่อสารในแบบจำลองโครงสร้างระบบสกาดาของวิทยานิพนธ์ฉบับนี้กำหนดให้มีข้อกำหนดในการสื่อสาร 2 ลักษณะคือ

- การสื่อสารระดับล่าง (Low-Level Communication Module) คือ การติดต่อสื่อสารระหว่าง RTU กับสถานีปฏิบัติการหลักโดยสื่อสารกันด้วยมาตรฐาน RS-485 โดยการโพลลิ่ง

- การสื่อสารระดับบน (High-level Communication Module) คือ การติดต่อสื่อสารระหว่างระบบลีนุกซ์ (Linux System) หรือสถานีหลัก (MTU) กับหน่วยระบบปฏิบัติการแบบผสมผสาน (Hybrid Operating System Module) โดยผ่านหน่วยบริการเครือข่าย (Server Module) โดยใช้โพรโตคอล TCP/IP เป็นมาตรฐานในการติดต่อสื่อสาร

นอกจากโครงสร้างทั้งสามหัวข้อที่กล่าวมาแล้ว ยังต้องการหน่วยปฏิบัติการอีกหน่วยหนึ่งที่ทำหน้าที่ในการประยุกต์ใช้งานของระบบฐานข้อมูลและแสดงผลในรูปของการประสานงานกับผู้ใช้เป็นหลักส่วนนั้นคือ

หน่วยระบบปฏิบัติการแบบผสมผสาน (Hybrid Operating System Module) เป็นส่วนของโปรแกรมการประยุกต์ใช้งานบนไคลเอนท์ที่สามารถทำงานของระบบสกาดาได้เกือบทุกอย่างยกเว้นเพียงการควบคุมการโพลลิ่งข้อมูลของ RTU เท่านั้นและยังทำงานได้ทั้งบนระบบปฏิบัติการที่เป็นไมโครซอฟต์วินโดว์และลีนุกซ์ โดยใช้ข้อกำหนดการสื่อสารระดับบนในการดึงข้อมูลจากฐานข้อมูลมาแสดงผลบนสื่อที่สามารถสื่อความหมายได้อย่างชัดเจน

3.2 คุณสมบัติของซอฟต์แวร์สกาดาบนระบบปฏิบัติการลีนุกซ์

1. แสดงผลการทำงานในขณะเวลาจริง (Real - Time Monitoring) สามารถตรวจสอบตามการทำงานของระบบสกาดาในรูปแบบของตัวอักษร กราฟ หรือกราฟฟิกอื่นๆที่สามารถสื่อความหมายได้อย่างชัดเจนในขณะที่เทียบกับเวลาจริง

2. เก็บบันทึกข้อมูลในรูปแบบเทียบกับเวลาจริง (Real - Time Database) สนับสนุนข้อมูลทั้งแบบต่อเนื่องและแบบไม่ต่อเนื่องรวมทั้งข้อมูลจำนวนเต็มและตัวอักษร

3. แสดงผลประวัติการทำงานในรูปแบบเวลาจริง (Real-Time and Historical Trends) สามารถดูการเปลี่ยนแปลงของค่าต่างๆ เทียบกับเวลาได้ และสามารถใช้เคอร์เซอร์อ่านค่า ณ จุดที่ต้องการได้

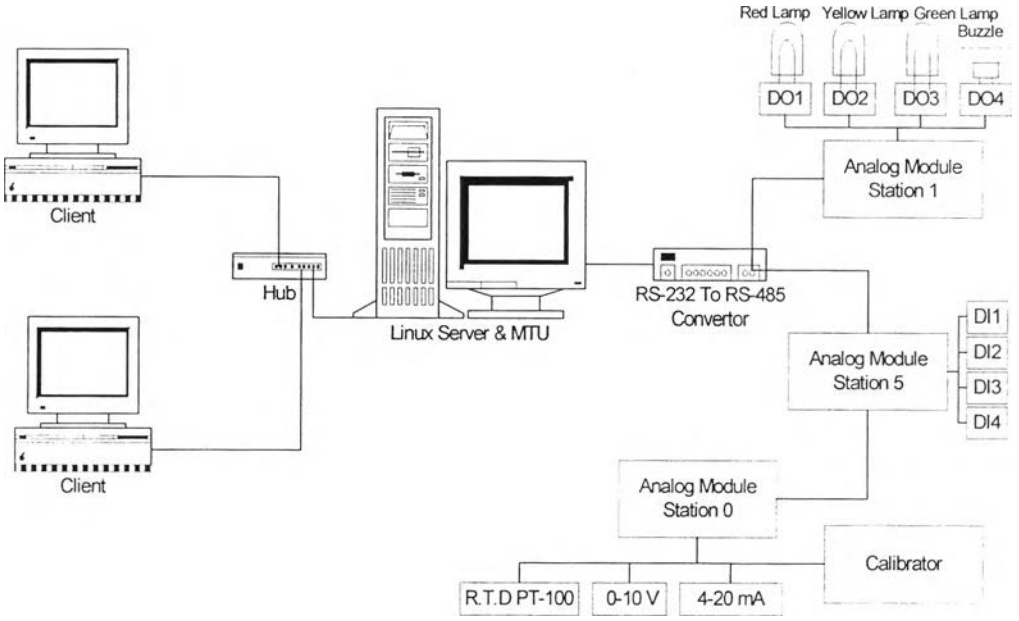
4. สามารถกำหนดเกณฑ์ในแจ้งความผิดพลาดของค่าข้อมูลที่รับเข้ามา (Alarm Capabilities) สัญญาณเตือนที่เกิดขึ้นสามารถแสดงบนหน้าจอในลักษณะการเปลี่ยนแปลงของสี

5. ระบบไคลเอนท์/เซิร์ฟเวอร์ (Client-Server System) ในระบบขนาดใหญ่ที่ซับซ้อนนั้น ระบบควบคุมแบบอัตโนมัติจะต้องเป็นระบบเครือข่าย การสนับสนุนระบบไคลเอนท์/เซิร์ฟเวอร์ทำให้สามารถกระจายงานต่างๆ ไปยังคอมพิวเตอร์แต่ละตัวในเนตเวิร์ก โดยอาศัยข้อมูลกลางร่วมกันได้ทั้งระบบปฏิบัติการที่เป็น ลีนุกซ์หรือไมโครซอฟต์วินโดว์

6. สนับสนุนการทำรายงาน (Reporting) รายงานประจำวันเกี่ยวกับข้อมูลต่างๆ ในรูปไฟล์เอกสารได้

3.3 การออกแบบโครงสร้างทางซอฟต์แวร์ของระบบสกาดา

เมื่อทำการกำหนดคุณสมบัติของส่วนต่างๆของโครงสร้างของระบบสกาดาบนระบบปฏิบัติการลีนุกซเรียบร้อยแล้ว เพื่อให้ง่ายต่อการสร้างซอฟต์แวร์ของระบบสกาดาได้ลองเขียนโครงสร้างทั้งหมดที่ออกแบบไว้ในรูปแบบที่สามารถนำไปทดสอบการใช้งานได้ออกมาดังแสดงในรูปที่ 3.4 ในขั้นตอนนี้จะทำให้สามารถมองระบบโดยภาพรวมได้ชัดเจนยิ่งขึ้นและจากการสังเกตโครงสร้างของระบบสกาดาที่ในรูปที่ 3.4 พบว่า การพัฒนาระบบควบคุมและรวบรวมข้อมูลสกาดาที่มีสถานีหลักเป็นแบบรวมฟังก์ชันไว้ที่ศูนย์กลางและมีการติดต่อกับสถานีปฏิบัติการซึ่งเป็นแหล่งรวม RTU โดยสื่อสารกันด้วยมาตรฐาน RS-485 โดยการโพลลิ่งและมีการกระจายข้อมูลในรูปของฐานข้อมูลเซิร์ฟเวอร์แล้วสามารถกล่าวได้ว่าจะมีองค์ประกอบสำคัญและรายละเอียดการทำงานดังต่อไปนี้



รูปที่ 3.4 โครงสร้างการติดตั้งระบบฮาร์ดแวร์เพื่อทดสอบการทำงานของระบบสกาดาบนระบบปฏิบัติการลีนุกซ์

3.3.1 องค์ประกอบที่สำคัญของซอฟต์แวร์สกาดา จากโครงสร้างการทำงานของสถานีหลักในรูปที่ 3.2 และโครงสร้างของระบบสื่อสารในรูปที่ 3.3 แล้ว ส่วนที่สำคัญของระบบสกาดาสามารถแยกออกได้เป็น 3 ส่วนคือการสื่อสารทางพอร์ตอนุกรมเพื่อติดต่อกับ RTU ของสถานีหลัก, การบันทึกข้อมูลลงในฐานข้อมูลและการแสดงผล ซึ่งมีรายละเอียดการสร้างซอฟต์แวร์มีดังต่อไปนี้

3.3.1.1 ส่วนติดต่อสื่อสารผ่านทางพอร์ตอนุกรม (Serial Port)

ในหัวข้อนี้จะกล่าวถึงสองส่วนสำคัญที่ใช้ในซอฟต์แวร์สกาดา คือหน่วยควบคุมการติดต่อสื่อสาร และ หน่วยควบคุมการโพลลิ่งข้อมูล ซึ่งใช้เป็นส่วนสำคัญสำหรับการติดต่อระดับล่างกับ RTU เพื่อที่จะควบคุมและรวบรวมข้อมูลต่างๆจาก RTU มาเก็บในฐานข้อมูลหรือแสดงผลในรูปแบบต่างๆ ดังอธิบายดังต่อไปนี้

หน่วยควบคุมการติดต่อสื่อสาร

ในการสื่อสารผ่านทางพอร์ตอนุกรมบนระบบปฏิบัติการลีนุกซ์ ก่อนเริ่มใช้งานเราจำเป็นต้องกำหนดคุณสมบัติพื้นฐานบางอย่างให้กับพอร์ตอนุกรม เช่น ความเร็วที่ใช้สำหรับการติดต่อ จำนวนบิตของข้อมูลรวมทั้งขนาดความจำชั่วคราวที่ใช้เป็นบัฟเฟอร์ รวมถึงลักษณะการติดต่อกับอาร์ดแวร์ของระบบปฏิบัติการลีนุกซ์ก็มีข้อกำหนดที่เป็นมาตรฐานที่ใช้ตามของ POSIX

(Portable Operating System Interface) เพื่อความเข้าใจโดยง่าย จะทำการกำหนดคุณสมบัติการเข้าไปควบคุมการสื่อสารผ่านพอร์ตอนุกรมเป็นลำดับขั้น ดังต่อไปนี้

1. การกำหนดชนิดอุปกรณ์ที่ใช้ติดต่อสื่อสารอนุกรม ถ้าต้องการจะติดต่อพอร์ตอนุกรมบนระบบปฏิบัติการลีนุกซ์นั้นมีข้อกำหนดที่ต้องรู้ก่อนว่า"ลีนุกซ์มองอุปกรณ์ฮาร์ดแวร์ในรูปแบบของระบบไฟล์ (Files System) และมีการเรียกการทำงานผ่านระบบไฟล์เช่นกัน" โดยถ้าต้องการจะติดต่อพอร์ต 1 จะต้องอ้างไปยังไฟล์ "/dev/ttys0" และพอร์ต 2 ที่ "/dev/ttys1"

2. การกำหนดความเร็วและคุณสมบัติอื่นๆที่ใช้ในการสื่อสาร ซึ่งเป็นการกำหนดคุณสมบัติให้กับพอร์ตสื่อสารตามข้อกำหนดของ POSIX สำหรับการสื่อสารแบบอนุกรม โดยจะกำหนดเข้าไปที่ตัวบ่งชี้ (Flag) ของโครงสร้างการกำหนดควบคุมเทอร์มินอลดังแสดงในตารางที่ 3.1 ซึ่งตัวบ่งชี้ C_Cflag จะเป็นตัวบอกได้ว่าสามารถกำหนดอะไรได้บ้างดังแสดงในตารางที่ 3.2

ตารางที่ 3.1 ตัวบ่งชี้ (flag) ในการกำหนดการทำงานให้กับพอร์ตอนุกรม

Member	Description
c_cflag	Control options
c_lflag	Line options
c_iflag	Input options
c_oflag	Output options
c_cc	Control characters
c_ispeed	Input baud (new interface)
c_ospeed	Output baud (new interface)

รูปแบบในการกำหนดความเร็วจะใช้คำสั่งต่อไปนี้

```
Var Comport : termios;
// กำหนดตัวแปร Comport ให้เป็นตัวแปรชนิด termios
Cfsetispeed (Comport,B19200)
// กำหนดความเร็วในการอ่านข้อมูลให้มีค่าเท่ากับ 19200 บิต/วินาที
Cfsetospeed (Comport,B19200)
// กำหนดความเร็วในการส่งข้อมูลให้มีค่าเท่ากับ 19200 บิต/วินาที
```

I 21234000

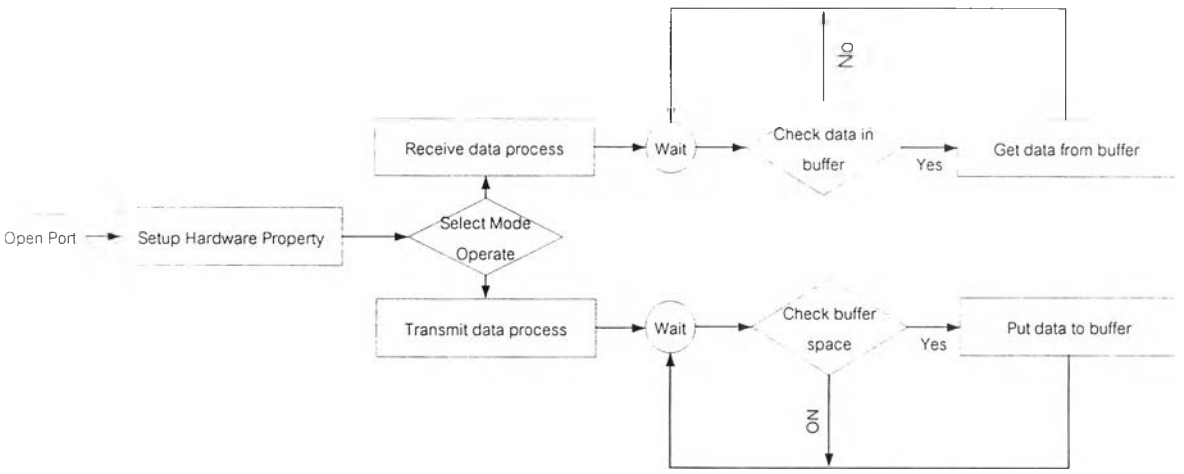
```
รูปแบบในการกำหนดการใช้จำนวนบิตในการรับส่งข้อมูล
    Comport.c_cflag := Comport.c_cflag and not CSTOPB
  // กำหนดให้มีการใช้ 1 บิตหยุด (Stop bit) ในการส่งข้อมูล
    Comport.c_cflag := Comport.c_cflag and not CSIZE
  // กำหนดให้ไม่มีการใช้บิตพราง (Mask bit) ในรูปแบบการส่งข้อมูล
    Comport.c_cflag := Comport.c_cflag or  CS8
  // กำหนดให้มีการส่งข้อมูลแบบ 8 บิต (ถ้า 7บิตก็จะเป็น CS7)
รูปแบบในการกำหนดการตรวจสอบพาริตีบิต
    Comport.c_cflag := Comport.c_cflag and (not PARENB)
   // กำหนดให้ไม่มีการตรวจสอบพาริตีบิต
    Comport.c_cflag := Comport.c_cflag or  PARENB
    Comport.c_cflag := Comport.c_cflag and (not PARODD)
  // กำหนดให้มีการตรวจสอบบิตคู่
    Comport.c_cflag := Comport.c_cflag or  PARENB
    Comport.c_cflag := Comport.c_cflag or  PARODD
  // กำหนดให้มีการตรวจสอบบิตคี่
```

นอกจากนี้แล้วยังต้องกำหนดลักษณะการควบคุมสายงาน (Flow Control) ว่าต้องการควบคุมสายงานแบบใดในรูปของซอฟต์แวร์อัตโนมัติหรือปล่อยให้เป็นไปตามการทำงานของระบบฮาร์ดแวร์ ส่วนการกำหนดรายละเอียดในการสร้างซอฟต์แวร์ที่นอกเหนือที่กล่าวไปแล้วนั้นเพื่อที่จะเข้าไปควบคุมพอร์ตอนุกรมให้มีความสามารถในลักษณะใดก็เป็นเทคนิคของผู้ออกแบบว่าจะใช้ข้อกำหนดที่มีให้ในรูปแบบใด

ตารางที่ 3.2 คุณสมบัติของตัวบ่งชี้ (flag) c_cflag [8]

Constant	Description
CBAUD	Bit mask for baud rate
B0	0 baud (drop DTR)
B50	50 baud
B75	75 baud
B110	110 baud
B134	134.5 baud
B150	150 baud
B200	200 baud
B300	300 baud
B600	600 baud
B1200	1200 baud
B1800	1800 baud
B2400	2400 baud
B4800	4800 baud
B9600	9600 baud
B19200	19200 baud
B38400	38400 baud
B57600	57,600 baud
B76800	76,800 baud
B115200	115,200 baud
EXTA	External rate clock
EXTB	External rate clock
CSIZE	Bit mask for data bits
CS5	5 data bits
CS6	6 data bits
CS7	7 data bits
CS8	8 data bits
CSTOPB	2 stop bits (1 otherwise)
CREAD	Enable receiver
PARENB	Enable parity bit
PARODD	Use odd parity instead of even
HUPCL	Hangup (drop DTR) on last close
CLOCAL	Local line – do not change "owner" of port
LOBLK	Block job control output
CNEW_RTSCTS CRTSCTS	Enable hardware flow control (not supported on all platforms)

หลังจากกำหนดคุณสมบัติของการสื่อสารทางพอร์ตอนุกรมเรียบแล้วขั้นตอนต่อไปที่จะต้องทำคือจะต้องกำหนดกิจกรรมที่สำหรับการรองรับการเรียกใช้พอร์ตอนุกรมในแต่ละครั้ง ดังอธิบายได้ตามรูปที่ 3.4



รูปที่ 3.4 ขั้นตอนของกิจกรรมเมื่อมีการเรียกใช้พอร์ตอนุกรม [17]

ในขั้นตอนแรกของการเรียกใช้พอร์ตอนุกรมนั้นจะเริ่มที่การตรวจสอบการเปิดหรือปิดพอร์ตก่อนแล้วจากนั้นก็จะกำหนดคุณสมบัติที่ต้องการให้พอร์ตอนุกรมทำงานได้ตามจุดประสงค์ลงไปตามลำดับที่กล่าวไปแล้วในหัวข้อก่อนหน้านี้หลังจากกำหนดคุณสมบัติเสร็จก็จะต้องทำการเลือกลักษณะการทำงานว่าขณะนี้ต้องการที่จะส่งข้อมูลหรือต้องการที่จะรับข้อมูล หลังจากนี้จะเป็นส่วนหน้าที่ของการทำงานของโปรแกรมคอยตรวจสอบว่าการทำงานตรงกับที่กำหนดไว้ตามคุณสมบัติหรือไม่ เช่น เลือกเหตุการณ์การทำงานแบบรับข้อมูลแบบอัตโนมัติ การทำงานของโปรแกรมต้องมีการตรวจสอบภายในบัฟเฟอร์ของพอร์ตอนุกรมตลอดเวลาว่ามีข้อมูลอยู่หรือไม่ ถ้ามีข้อมูลอยู่ก็จะสร้างสัญญาณอินเตอร์รัพต์ไปบอกโปรแกรมหลักที่ทำงานอยู่ว่ามีข้อมูลมาแล้วพร้อมที่จะรับหรือไม่ เป็นต้น

หน่วยควบคุมการโพลลิ่งข้อมูล (Polling Data Control)

เมื่อได้อุปกรณ์ที่ใช้ควบคุมการการสื่อสารแล้ว ในขั้นตอนต่อไปจะต้องทำการออกแบบโครงสร้างที่จะใช้งานกับการสื่อสารแบบมอดบัส (Modbus) บนมาตรฐานของ RS-485 ดังนั้นต้องกำหนดคุณสมบัติของโปรแกรมเสียก่อนว่าในระบบจะต้องมีการทำงานอะไรบ้าง ซึ่งเรียงลำดับตามจุดประสงค์ดังต่อไปนี้

1. ต้องสามารถทำการรวบรวมข้อมูลจาก RTU ได้อย่างต่อเนื่องและครบทุก RTU ภายใต้เวลาที่กำหนดไว้อย่างถูกต้อง ถ้าที่สถานีต่ออยู่นั้นไม่เกิดความผิดปกติในการสื่อสาร

2. ต้องสามารถตรวจสอบสถานะการสื่อสารระหว่างสถานีหลักกับ RTU ได้ว่ามีการสื่อสารอยู่ในสภาวะปกติหรือเปล่าถ้าไม่ปกติต้องสามารถแจ้งเตือนได้

3. ต้องสามารถแยกแยะและตรวจสอบข้อมูลที่รวบรวมเข้ามาได้อย่างถูกต้อง

จากคุณสมบัติตามที่กล่าวมาข้างต้น สามารถกำหนดได้ว่าในโปรแกรมที่จะสร้างขึ้นสำหรับการโพลลิ่งข้อมูลนั้นจะต้องประกอบไปด้วย 2 หน่วยที่สำคัญคือ

- หน่วยเก็บคุณสมบัติของ RTU มีหน้าที่ในการเก็บข้อมูลต่อไปนี้
    - หมายเลขเครื่อง (Station Number)
    - ประเภทของข้อมูลประเภทแอนะล็อก (Analog Type)
    - ข้อมูลชนิดแอนะล็อกอินพุต (Analog Data Input)
    - ข้อมูลชนิดดิจิตอลอินพุต (Digital Input)
    - ข้อมูลชนิดดิจิตอลเอาต์พุต (Digital Output)
    - ข้อมูลบอกสถานะการติดต่อของสถานี (Station Connect)

  โดยข้อมูลเหล่านี้ควรจะถูกกำหนดในรูปของคลาส (Class) หรือแผ่นตารางทำการ (Spread Sheet) เพื่อที่สะดวกต่อการนำไปใช้งาน

- หน่วยการโพลลิ่งข้อมูล (Polling Data) มีหน้าที่ในการดึงข้อมูลจาก RTU โดยใช้การควบคุมพอร์ตสื่อสารเข้ามาช่วยเพื่อดึงข้อมูลเข้ามาเก็บไว้ในหน่วยเก็บคุณสมบัติของสถานีปลายทาง นอกจากนั้นแล้วในหน่วยนี้ต้องมีการกำหนดประเภทความผิดพลาดที่อาจเกิดขึ้นได้พร้อมทั้งกำหนดรหัสการทำงานเพื่อเป็นเกณฑ์ไว้สำหรับบันทึกในหน่วยเก็บคุณสมบัติของ RTU และฐานข้อมูลได้ เช่น

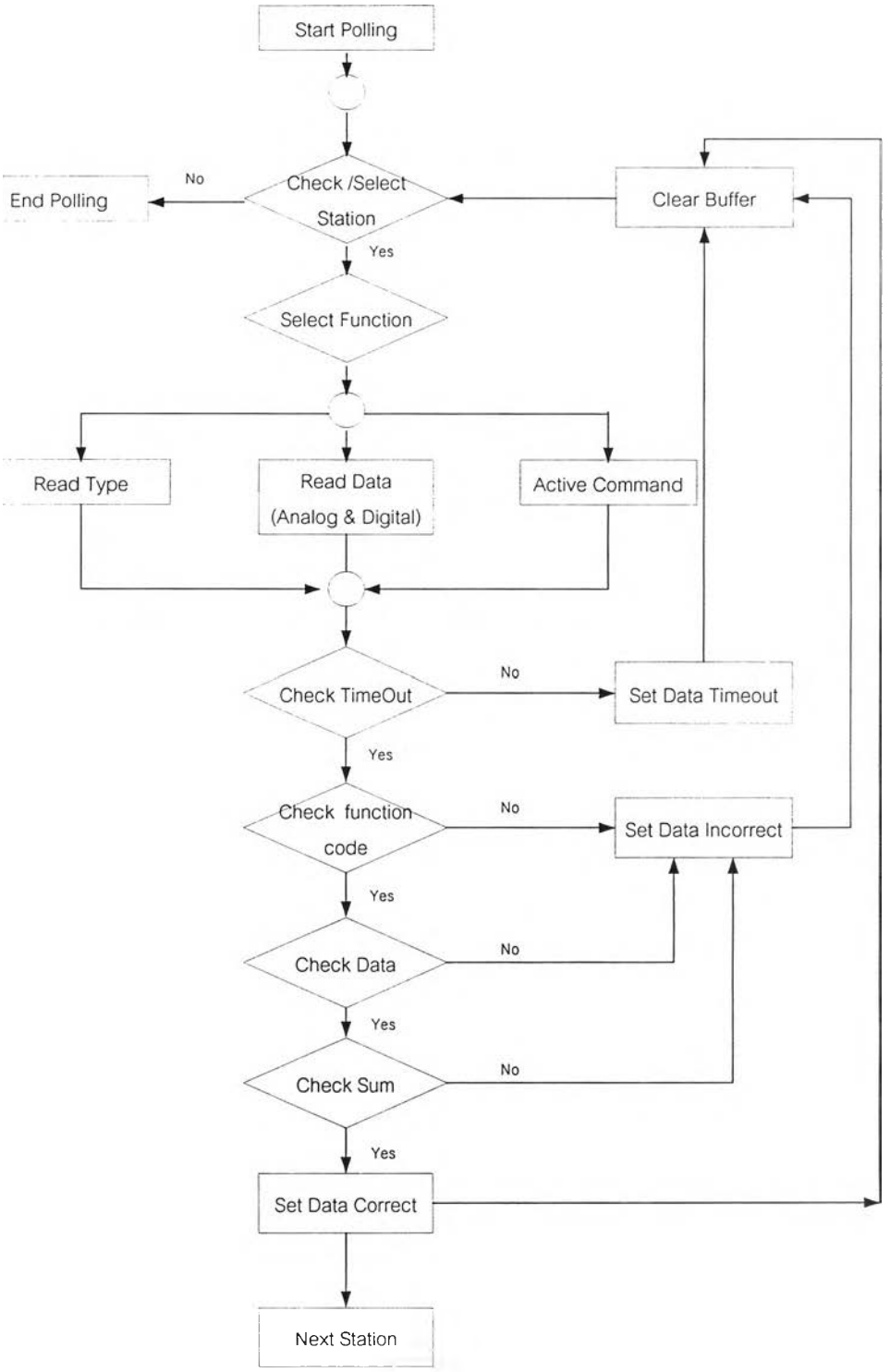
ตารางที่ 3.3 การกำหนดรหัสที่ใช้บอกสถานะของข้อมูล

Function Type	Code
TimeOut	T
Not Connected	NC
Connected	C
Over Range	O
Under Range	U
Mis Data	X
Not Use	N

เมื่อกำหนดคุณสมบัติเรียบร้อยแล้วก็คงเหลือในส่วนโครงสร้างและกิจกรรมของการโพลลิ่งโดยสามารถที่จะเรียงลำดับการทำงานได้ดังนี้

- ตรวจสอบจำนวนสถานีที่ต้องทำการโพลลิ่ง ถ้าไม่มีให้หยุดการทำงาน
- ตรวจสอบชุดคำสั่งว่าในการโพลลิ่งครั้งนี้ต้องการข้อมูลประเภทใด เช่น ต้องการประเภทของอุปกรณ์ที่ต่ออยู่ หรือต้องการอ่านข้อมูลชนิดแอนะล็อก เป็นต้น
- อ่านข้อมูลที่ต้องการกลับมา
- ตั้งเวลารอคอยในการรอสัญญาณในการอ่านข้อมูลกลับ ถ้าครบกำหนดเวลาแล้วไม่มีการส่งข้อมูลจาก RTU ให้ทำการข้ามไปทำงานในสถานีถัดไปแล้วบันทึกชนิดของความผิดพลาดประเภท 'T' (TimeOut) ลงไปที่หน่วยเก็บคุณสมบัติของ RTU
- ตรวจสอบความถูกต้องของข้อมูลที่ส่งมา ถ้าข้อมูลไม่เกิดความผิดพลาดให้ทำการบันทึกข้อมูลลงไปในหน่วยเก็บคุณสมบัติของ RTU แต่ถ้าเกิดความผิดพลาดในเงื่อนไขใดก็ให้บันทึกชนิดของความผิดพลาดลงไปแทน แล้วทำการโพลลิ่งข้อมูลในสถานีถัดไปจนครบจำนวน

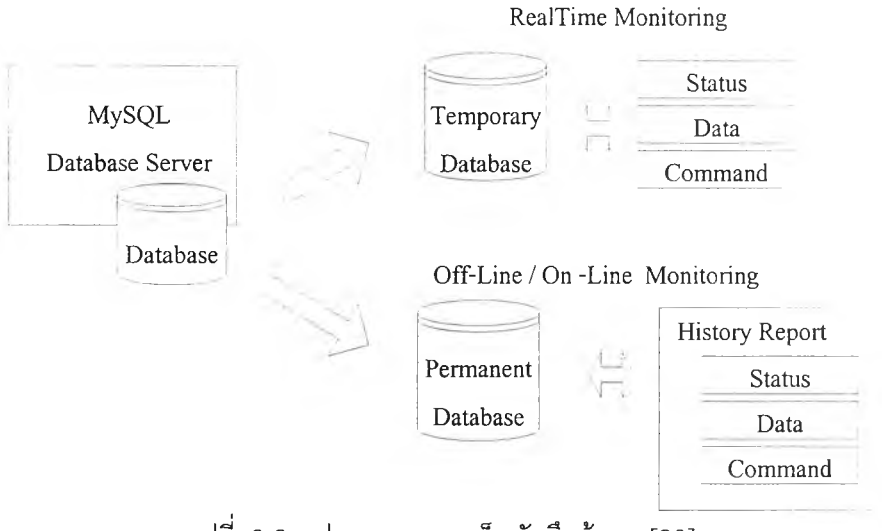
จากลำดับการทำงานดังกล่าว สามารถแสดงเป็นโครงสร้างการทำงานดังรูปที่ 3.5



รูปที่ 3.5 โครงสร้างการโพลลิ่งข้อมูล

3.3.1.2 ส่วนการทำงานกับฐานข้อมูล (Database)

ในวิทยานิพนธ์ฉบับนี้ได้ออกแบบระบบฐานข้อมูลไว้เป็นประเภทไคลเอนท์/เซิร์ฟเวอร์เป็นลักษณะการเข้าถึงข้อมูลแบบทูเทีย (Two-tiered) ซึ่งจะเน้นการเข้าถึงฐานข้อมูลแบบมัลติยูสเซอร์ (Multi - User) บนฐานข้อมูลสัมพันธ์ (Relation Database Management System) หรือฐานข้อมูลเซิร์ฟเวอร์ (Database Server) กล่าวคือผู้ใช้สามารถทำงานกับฐานข้อมูลพร้อมๆกันได้หลายคนโดยเรียกผ่านทางการประยุกต์ใช้งานผ่านเครื่องไคลเอนท์ (Client Application) โดยโครงสร้างฐานข้อมูลที่ใช้ในระบบควบคุมและรวบรวมข้อมูลสกาดาจะถูกแบ่งตามหน้าที่การทำงานได้เป็น 2 ประเภทคือ



รูปที่ 3.6 รูปแบบของการเก็บบันทึกข้อมูล [20]

1. ฐานข้อมูลประเภทชั่วคราว (Temporary Database) ข้อมูลที่อยู่ในฐานข้อมูลชนิดนี้จะมีอายุการใช้งานเพียงชั่วคราวเท่านั้น เพื่อจะส่งข้อมูลต่อให้กับส่วนแสดงผลในรูปเทียบกับเวลาจริง (Real-Time) เป็นหลัก และแต่ละชนิดของข้อมูลก็มีอายุใช้การงานไม่เท่ากัน เช่น ข้อมูลชนิดที่แสดงสถานะการติดต่อกับ RTU จะคงอยู่เฉพาะตอนเรียกตรวจสอบประเภทของข้อมูลที่ได้รับเข้ามา หลังจากเรียกตรวจสอบใหม่ก็จะเปลี่ยนแปลงไปตามค่าที่เรากำหนดไว้ ส่วนคำสั่ง (Command) ก็จะถูกลบทันทีที่สถานีปฏิบัติงานตามคำสั่งเสร็จสิ้น เป็นต้น

2. ฐานข้อมูลประเภทถาวร (Permanent Database) จะทำหน้าที่ในการบันทึกข้อมูลทุกประเภทไว้ในฐานข้อมูลประวัติโดยจะทำการบันทึกข้อมูลไปพร้อมกับฐานข้อมูล

ประเภทชั่วคราว (Temporary Database) และจะมีการสำรองข้อมูลในรูปของไฟล์เอกสารตามเวลาที่กำหนดไว้ โดยหน้าที่หลักของฐานข้อมูลชนิดนี้คือช่วยในการดูประวัติย้อนหลังในการทำงานและสนับสนุนการทำรายงานรวมทั้งการป้องกันข้อมูลสูญหายถ้าระบบฐานข้อมูลประเภทชั่วคราวเกิดการขัดข้อง

โครงสร้างของฐานข้อมูลสองชนิดนี้ได้แสดงไว้ดัง รูปที่ 3.6 โดยที่ข้อมูลที่เก็บไว้ในฐานข้อมูลทั้งสองแบบจะเป็นข้อมูลประเภทเดียวกันซึ่งเป็นข้อมูลที่แสดงถึง

- **ฐานข้อมูลที่เก็บข้อมูลประเภทของช่องสัญญาณที่ใช้วัด** (แรงดัน, กระแส ฯ) ซึ่งกำหนดไว้ในฐานข้อมูลชื่อสถานะ (STATUS) มีรายละเอียดดังแสดงในตารางที่ 3.4 สิ่งสำคัญที่ต้องกำหนดเมื่อทราบประเภทของอุปกรณ์ที่วัดแล้วคือ ค่าสัมประสิทธิ์ของตัวคูณที่ใช้แสดงผล เหตุผลที่ต้องกำหนดก็เพราะเพื่อความสะดวกในการสื่อสารและนำไปใช้งาน เช่น เมื่อข้อมูลที่รับเข้ามาอยู่ในรูปของสัญญาณกระแสช่วง 4-20 มิลลิแอมป์ แต่ต้องการนำไปใช้งานในรูปของอุณหภูมิ 0-200 องศา การเข้าไปแก้ไขข้อมูลดิบที่รับเข้ามาเป็นเรื่องที่ไม่ควรทำเพราะอาจสูญเสียข้อมูลที่สำคัญได้ ดังนั้นจึงต้องไปกำหนดที่ค่าของสัมประสิทธิ์ของตัวคูณแทนเพื่อรักษาข้อมูลเดิมไว้และสะดวกที่จะเปลี่ยนแปลงอีกบ่อยครั้ง นอกจากนั้นเรายังต้องกำหนดรหัสของประเภทสัญญาณไว้เป็นข้อกำหนดในการวัดอีกด้วย ดังแสดงในตารางที่ 3.5

ตารางที่ 3.4 **รายละเอียดของฐานข้อมูลที่แสดงประเภทของช่องสัญญาณที่ใช้วัด**

Data Type	Name Field	Data Type	Name Field
Name Station	Station_Type	Analog Type 8	AT8
Datetime	Datetime	Coefficient Of AT1	COF 1
Analog Type 1	AT1	Coefficient Of AT2	COF 2
Analog Type 2	AT2	Coefficient Of AT3	COF 3
Analog Type 3	AT3	Coefficient Of AT4	COF 4
Analog Type 4	AT4	Coefficient Of AT5	COF 5
Analog Type 5	AT5	Coefficient Of AT6	COF 6
Analog Type 6	AT6	Coefficient Of AT7	COF 7
Analog Type 7	AT7	Coefficient Of AT8	COF 8

ตารางที่ 3.5 รายละเอียดของการกำหนดรหัสการทำงาน [22]

รหัส	ชนิดของสัญญาณ		ค่าวัด (Floating Point)	ค่าที่ได้รับ (Sign Integer)	ตัวหารกลับ
00	Not Use		-	-	-
01	Thermocouple	R	0 - 1700°C	0-1700	1
02		S	0 - 1700°C	0-1700	1
03		K	(-)250.0 - 1300.0°C	(-)2500 – 13000	10
04		E	0.0 - 1000.0°C	0 – 10000	10
05		J	(-)200.0 - 700.0°C	(-)2000 - 7000	10
06		T	(-)250.0 - 400.0°C	(-)2500 – 4000	10
07		B	0 - 1800°C	0 1800	1
08	R.T.D. Pt100		(-)200.0 - 800.0°C	(-)2000-8000	10
09	Voltage(mV) 0 - 100		0.00 - 100.00 mV	0 – 10000	100
10	Voltage (V)	0 - 5	0.000 - 5.000 V	0 – 5000	1000
11		0 - 10	0.000 - 10.000 V	0 – 10000	1000
12	Current (mA)	0 - 20	0.00 - 20.00 mA	0 – 2000	100
13		0 - 40	0.00 - 40.00 mA	0 – 4000	100

- **ฐานข้อมูลที่เก็บข้อมูลของชุดคำสั่งในการควบคุม** RTU ซึ่งกำหนดไว้ในฐานข้อมูลชื่อคำสั่ง (Command) มีรายละเอียดดังแสดงในตารางที่ 3.5

ตารางที่ 3.6 รายละเอียดของฐานข้อมูลที่แสดงชุดคำสั่งในการควบคุมสถานีปลายทาง

Data Type	Name Field
Command Index	Id_Command
Name Station	Station_Command
Command Status	Status_C
Datetime	Datatime
Select Channel Active	Sel
Active Digital Output Ch 1	Active_Do 1
Active Digital Output Ch 1	Active_Do 2
Active Digital Output Ch 1	Active_Do 3
Active Digital Output Ch 1	Active_Do 4

- **ฐานข้อมูลที่เก็บข้อมูลค่าจากการวัดของ RTU** คือเก็บข้อมูลชนิดแอนะล็อกอินพุต, ดิจิตอลอินพุต, ดิจิตอลเอาต์พุต และรหัสความผิดพลาดประเภทต่างๆ เช่น 'T' (TimeOut) เป็นต้น มีรายละเอียดดังแสดงในตารางที่ 3.7

ตารางที่ 3.7 รายละเอียดของฐานข้อมูลที่แสดงฐานข้อมูลที่เก็บข้อมูลค่าจากการวัดของ RTU

Data Type	Name Field	Data Type	Name Field
Data Index	ID	Digital Input 1	DI1
Name Station	Station_Num	Digital Input 2	DI2
Datetime	Datetime	Digital Input 3	DI3
Analog Input 1	A1	Digital Input 4	DI4
Analog Input 2	A2	Digital Output 1	DO1
Analog Input 3	A3	Digital Output 2	DO2
Analog Input 4	A4	Digital Output 3	DO3
Analog Input 5	A5	Digital Output 4	DO4
Analog Input 6	A6		
Analog Input 7	A7		
Analog Input 8	A8		

นอกจากนี้ยังสามารถเพิ่มฐานข้อมูลประเภทต่างๆเข้าไปอีกได้ขึ้นอยู่ความเหมาะสมกับประเภทงานที่นำไปใช้

3.3.1.3 **ส่วนการติดต่อกับผู้ใช้** (User Interface)

สามารถแบ่งได้เป็น 2 ส่วนใหญ่คือในฝั่งไคลเอนท์กับเซิร์ฟเวอร์โดยที่ในฝั่งของเซิร์ฟเวอร์อาจมีส่วนของการประยุกต์ใช้งานบนไคลเอนท์ด้วยก็ได้ ในโครงสร้างรายละเอียดการออกแบบจะแบ่งการทำงานตามลักษณะการทำงานกับฐานข้อมูลเป็นหลักคือ ส่วนที่ฮาร์ดแวร์ติดต่อกันโดยตรงกับฐานข้อมูลซึ่งเป็นส่วนสำคัญกับส่วนติดต่อกับผู้ใช้ของเซิร์ฟเวอร์และอีกส่วนหนึ่งคือส่วนที่ผู้ใช้ติดต่อกับฐานข้อมูลโดยตรงหรือเป็นส่วนที่เรียกว่าส่วนติดต่อผู้ใช้ของไคลเอนท์ มีรายละเอียดดังต่อไปนี้

ส่วนติดต่อกับผู้ใช้ของเซิร์ฟเวอร์ จะทำหน้าที่ในการตรวจสอบการติดต่อและการกำหนดค่าต่างๆให้ RTU รวมทั้งแสดงสถานะความเปลี่ยนแปลงทุกอย่างที่เกิดขึ้นขณะบันทึกข้อมูลลงฐานข้อมูล มีฟังก์ชันการทำงานดังต่อไปนี้

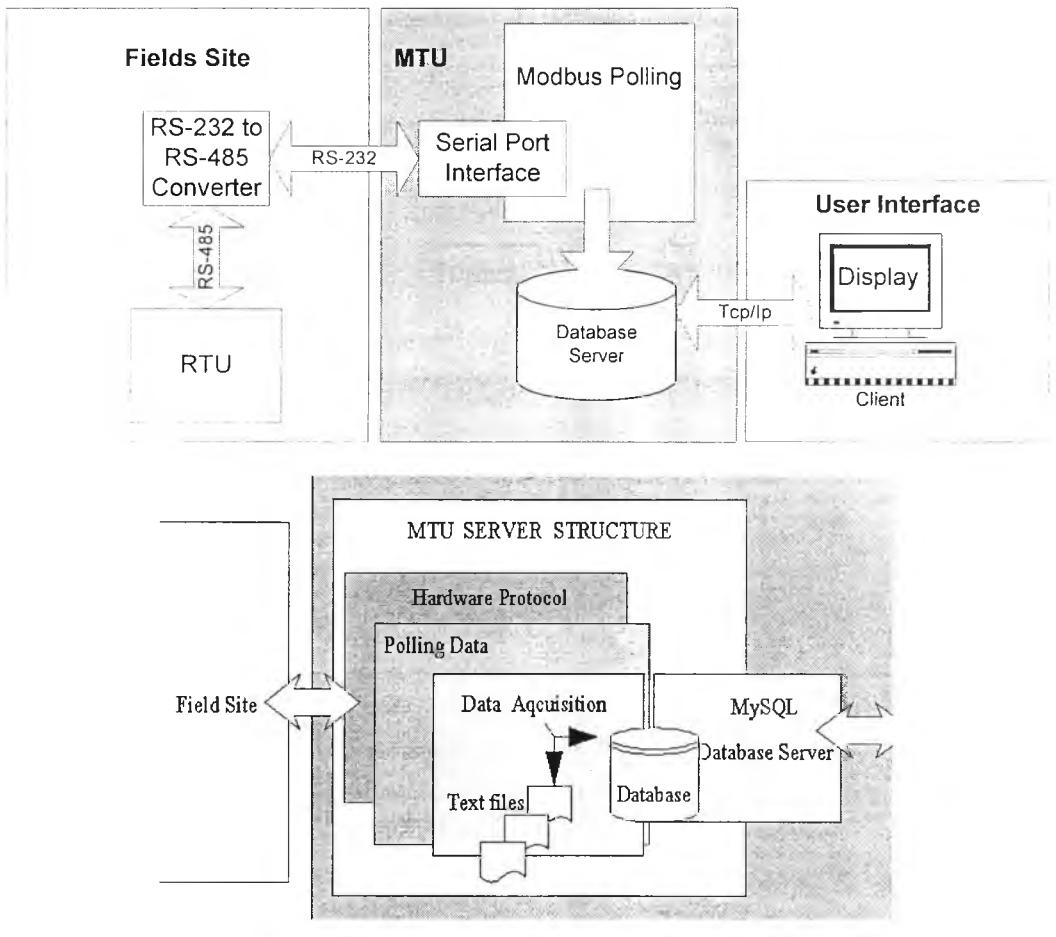
- ฟังก์ชันการตรวจสอบการติดต่อกับฐานข้อมูล
- ฟังก์ชันการตรวจสอบการติดตั้งของ RTU
- ฟังก์ชันการแยกประเภทของช่องสัญญาณที่วัด
- ฟังก์ชันการตั้งรอบเวลาในการเก็บข้อมูล
- ฟังก์ชันการควบคุม RTU
- ฟังก์ชันการควบคุมการแสดงผล

ส่วนติดต่อกับผู้ใช้ของไคลเอนท์ จะทำหน้าที่ในการดึงข้อมูลจากฐานข้อมูลมาแสดงผลในรูปกราฟฟิกรวมถึงรอรับคำสั่งจากผู้ใช้ดังนั้นการทำงานของส่วนติดต่อกับผู้ใช้ของไคลเอนท์ถือได้ว่าเป็นการทำงานที่ผู้ใช้ติดต่อกับฐานข้อมูลโดยตรง มีฟังก์ชันการทำงานดังต่อไปนี้

- ฟังก์ชันการตรวจรหัสผ่านของผู้ใช้
- ฟังก์ชันการตรวจสอบการติดต่อกับฐานข้อมูล
- ฟังก์ชันการตรวจสอบการติดตั้งของ RTU
- ฟังก์ชันการแยกประเภทของช่องสัญญาณที่วัด
- ฟังก์ชันการตั้งรอบเวลาในการแสดงผล
- ฟังก์ชันการควบคุม RTU
- ฟังก์ชันการควบคุมการแสดงผลในรูปแบบทั่วไป
- ฟังก์ชันการควบคุมการแสดงผลในรูปแบบของกราฟ
- ฟังก์ชันการควบคุมการแสดงผลจากฐานข้อมูลประวัติ

ในฟังก์ชันการทำงานของส่วนประสานงานกับผู้ใช้โดยความจริงแล้วเป็นส่วนที่มีความยืดหยุ่นสูงมากมีการเปลี่ยนแปลงบ่อย ขึ้นกับผู้พัฒนาในขณะนั้นเป็นใคร ถ้าต้องการที่ให้มีการพัฒนาได้อย่างต่อเนื่องแล้วควรจะแยกการทำงานของแต่ละฟังก์ชันออกเป็นหน่วย เพื่อความสะดวกต่อการแก้ไขและเปลี่ยนแปลง ในวิทยานิพนธ์ฉบับนี้ได้ออกแบบการทำงานของฟังก์ชันตามขอบเขตของระบบควบคุมและรวบรวมข้อมูลสกาดาขนาดกลางเท่านั้น ดังนั้นถ้าฟังก์ชันการทำงานอาจจะไม่ครอบคลุมทั้งหมดของระบบการทำงาน ก็สามารถเพิ่ม แก้ไข หรือตัดทิ้งออกไป ส่วนการอธิบายขบวนการเรียกใช้งานของแต่ละฟังก์ชันได้อธิบายไว้แล้วในหัวข้อถัดไป

3.3.2 ผังและโครงสร้างการทำงานของซอฟต์แวร์สกาดาบนระบบปฏิบัติการลีนุกซ์



รูปที่ 3.7 รูปแบบการทำงานอย่างง่ายของระบบสกาดา

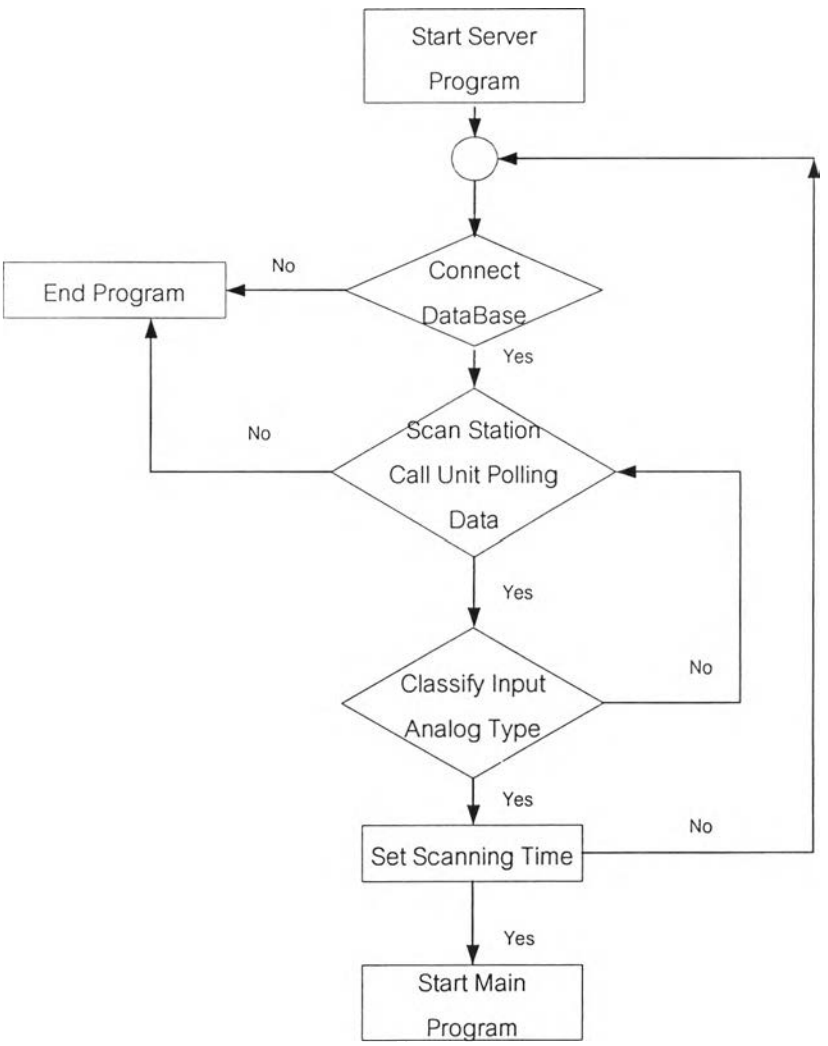
ในส่วนนี้จะเป็นการสรุปถึงขั้นตอนการทำงานอย่างคร่าวๆอีกครั้งหนึ่งว่าขั้นแรกของการทำงานจะเริ่มที่ระบบเซิร์ฟเวอร์โดยสถานีหลักมีการสั่งให้อ่านข้อมูลจาก RTU ทุกตัวที่ทำงานอยู่กับระบบสกาดาโดยการโพลลิ่งครั้งละ 1 RTU หลังจากนั้นนำข้อมูลที่ได้มาทำการตรวจสอบความถูกต้องและบันทึกลงในฐานข้อมูลชนิดต่างๆแล้ววนกับไปทำงานในขั้นตอนแรกอีกจนกว่าจะมีคำสั่งให้หยุดการทำงาน ส่วนในการทำงานของการประยุกต์ใช้งานบนไคลเอนท์นั้นจะทำหน้าที่ในการอ่านข้อมูลจากฐานข้อมูลประเภทต่างๆขึ้นมาแสดงผลตามฟังก์ชันการใช้งานที่มีการออกแบบไว้

นอกจากนี้แล้วยังสามารถสรุปรายละเอียดของการทำงานของซอฟต์แวร์สกาดาที่ออกแบบไว้ได้เป็น 2 ส่วนใหญ่ คือในฝั่งไคลเอนท์กับฝั่งเซิร์ฟเวอร์ ซึ่งมีโครงสร้างการทำงานดังต่อไปนี้

โครงสร้างการทำงานของส่วนติดต่อกับผู้ใช้ของส่วนการประยุกต์ใช้งานบนเซิร์ฟเวอร์

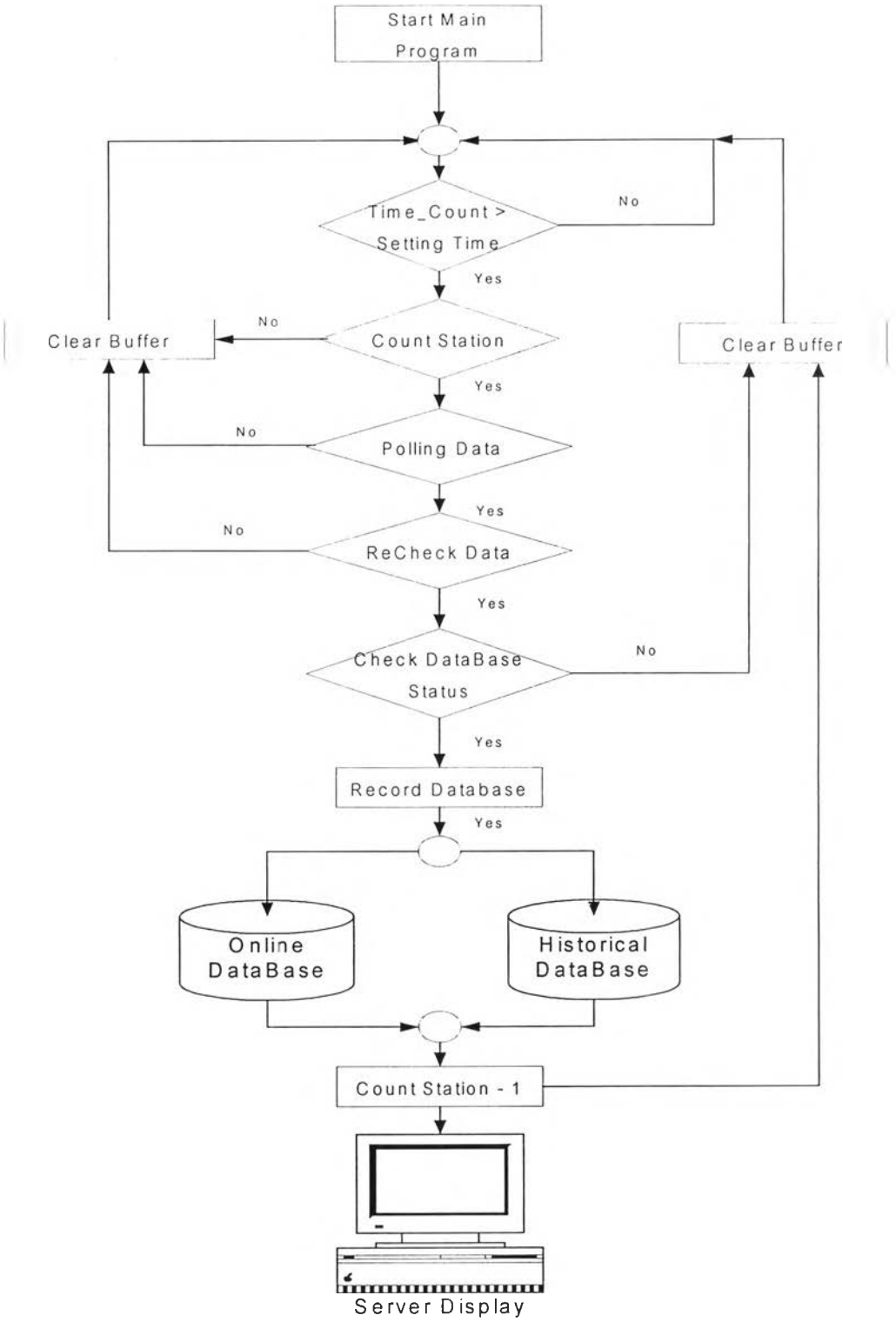
ในส่วนการประยุกต์ใช้งานบนเซิร์ฟเวอร์จะมีหน้าที่ในการทำงานอยู่ 2 อย่างคือ

1. ทำหน้าที่ในการกำหนดค่าเริ่มต้นให้กับระบบสกาดาเช่น การตรวจจำนวน RTU ที่ติดตั้งอยู่ หรือการตรวจสอบความพร้อมของฐานข้อมูลก่อนที่จะเริ่มทำงานดังแสดงได้ดังรูปที่ 3.8



รูปที่ 3.8 โครงสร้างเริ่มต้นการทำงานของส่วนการประยุกต์ใช้งานบนเซิร์ฟเวอร์

2. ทำหน้าที่ในการควบคุมการอ่านข้อมูลโดยดารโพลลิ่งจาก RTU ที่ติดตั้งอยู่ และทำการตรวจสอบข้อมูลที่อ่านเข้ามาเพื่อทำการบันทึกลงในฐานข้อมูลพร้อมทั้งรายงานผล ดังแสดงในรูปที่ 3.9

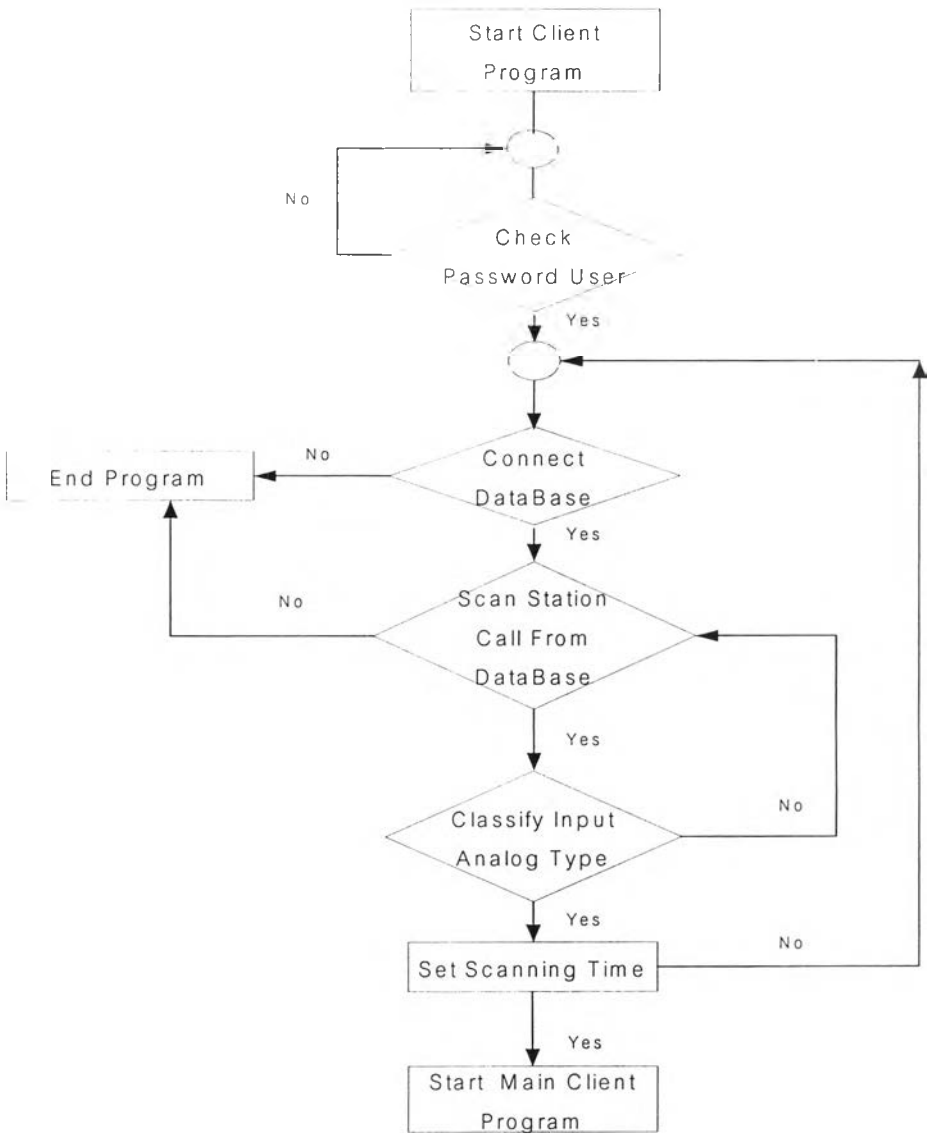


รูปที่ 3.9 โครงสร้างขณะที่ทำงานของส่วนการประยุกต์ใช้งานบนเซิร์ฟเวอร์

โครงสร้างการทำงานของส่วนติดต่อกับผู้ใช้ของส่วนการประยุกต์ใช้งานบนไคลเอนท์

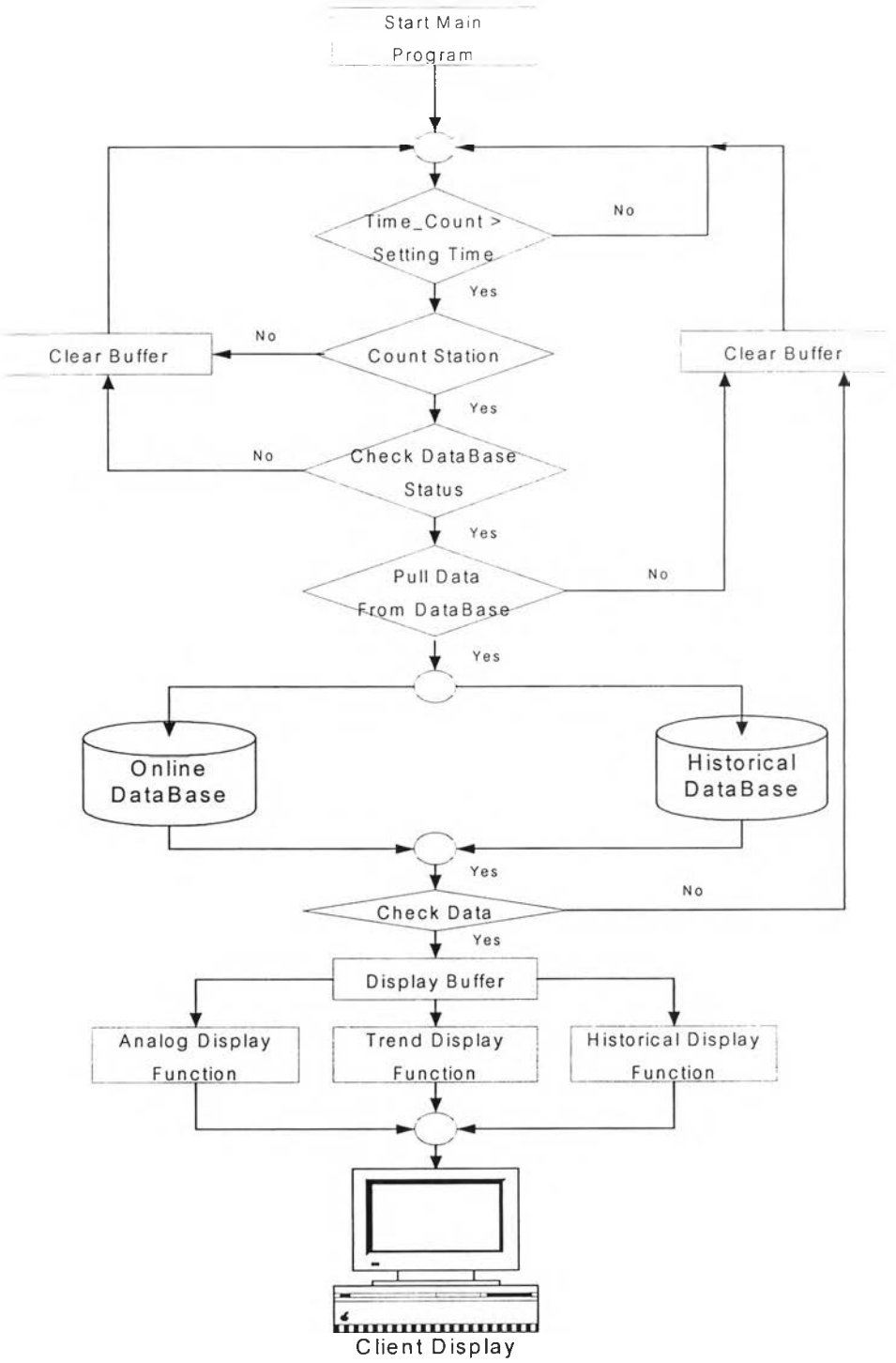
ในส่วนการประยุกต์ใช้งานบนไคลเอนท์จะมีหน้าที่ในการทำงานอยู่ 2 อย่างคือ

1. ทำหน้าที่ในการกำหนดค่าเริ่มต้นในการแสดงผลเช่น การตรวจสอบความพร้อมของฐานข้อมูลก่อนที่จะเริ่มทำงาน หรือการตรวจสอบความถูกต้องของชนิดของช่องสัญญาณที่ต้องการนำมาแสดง โดยในส่วนนี้จะเป็นการตรวจสอบผ่านระบบไคลเอนท์/เซิร์ฟเวอร์ ดังแสดงขั้นตอนการทำงานได้ดังรูปที่ 3.10



รูปที่ 3.10 โครงสร้างเริ่มต้นการทำงานของส่วนการประยุกต์ใช้งานบนไคลเอนท์

2. ทำหน้าที่ในการควบคุมและประสานงานกับผู้ใช้ในการแสดงผลทั้งในรูปแบบขณะเวลาจริงและฐานประวัติของข้อมูล โดยอาศัยหลักการทำงานของระบบไคลเอนท์/เซิร์ฟเวอร์ดังแสดงขั้นตอนการทำงานได้ดังรูปที่ 3.11



รูปที่ 3.11 โครงสร้างขณะทำงานของการประยุกต์ใช้งานบนไคลเอนท์

3.4 เครื่องมือสำหรับการออกแบบซอฟต์แวร์สกาดาบนระบบปฏิบัติการลีนุกซ์

อุปกรณ์ที่ใช้ศึกษาและพัฒนา

ซอฟต์แวร์

- โปรแกรมไคลิกซ์เวอร์ชัน 3 (Kylix 3) โปรแกรมเชิงภาษาภาพที่ใช้งานบนระบบปฏิบัติการลีนุกซ์ [13],[19]
- โปรแกรมเดลไฟล์ เวอร์ชัน 6 (Delphi 6)
- ตัวบริการฐานข้อมูลมายเอสคิวเอล (MySQL Database Server)
- ระบบปฏิบัติการ REDHAT 7.2 (Linux Operating System)
- ระบบปฏิบัติการไมโครซอฟต์วินโดว์ 98 , 2000 (Win98 , Win2000)

ฮาร์ดแวร์

- ตัวรับสัญญาณแอนะล็อก(Analog Input Module AI 210)
- ตัวแปลงสัญญาณจากมาตรฐาน RS – 232 เป็น RS – 485

สรุปท้ายบท

จากการออกแบบระบบสกาดาขนาดกลางแบบสถานีหลักเป็นแบบรวมฟังก์ชันไว้ที่ศูนย์กลางและมีการติดต่อกับแหล่งรวม RTU โดยสื่อสารกันด้วยมาตรฐาน RS-485 โดยการโพลลิ่งด้วยโพรโตคอลสื่อสารแบบ MODBUS ซึ่งสามารถตรวจสอบสถานะการทำงานได้โดยผ่านระบบปฏิบัติการไมโครซอฟต์วินโดว์และลีนุกซ์ สรุปได้ว่าโครงสร้างของระบบจะต้องประกอบไปด้วย 3 ส่วนที่สำคัญคือ

1. ส่วนติดต่อสื่อสารผ่านทางพอร์ตอนุกรม มีหน้าที่ในการอ่านข้อมูลจาก RTU ขึ้นมาเพื่อตรวจสอบและเก็บข้อมูลในฐานข้อมูลแบบต่างๆที่กำหนดไว้ ระบบสกาดาที่ออกแบบไว้สามารถเปลี่ยนไปใช้โพรโตคอลอื่นได้โดยแก้ไขที่ส่วนการโพลลิ่งข้อมูลดังแสดงตัวอย่างไว้ภายในบทและภาคผนวก ก
2. ส่วนการทำงานกับฐานข้อมูล เป็นส่วนที่นำข้อมูลที่เก็บไว้ออกเผยแพร่ทั้งในรูปแบบเทียบกับเวลาจริงและฐานข้อมูลประวัติโดยอาสัยหลักการทำงานแบบระบบไคลเอนท์/เซิร์ฟเวอร์
3. ส่วนการติดต่อกับผู้ใช้ เป็นส่วนบรรจุฟังชันการซึ่งมีหน้าที่ติดต่อประสานงานกับผู้ใช้

นอกจากนั้นแล้วผลจากการออกแบบยังได้ฟังก์ชันและคุณสมบัติต่างๆของซอฟต์แวร์สกาดาซึ่งจะอธิบายการใช้งานและการทดสอบในบทต่อไป