ระบบควบคุมแบบดิจิตอล
DIGITAL CONTROL SYSTEM

โดย

นายฉัตรชัย วาจาเกียรติ

นายวิทยา รุ่งเจริญวัฒนา

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตร์บัณฑิต

ภาควิชาวิศวกรรมระบบควบคุม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2534

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

007713

ปริญญานิพนธ์ปีการศึกษา 2534

ภาควิชา วิศวกรรมระบบควบคุม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้า เจ้าคุณทหารลาดกระบัง

เรื่อง ระบบควบคุมแบบดิจิตอล

ผู้จัดทำ 1. นายฉัตรชัย วาจาเกียรติ 311048

2. นายวิทยา รุ่งเจริญวัฒนา 311249

..... อาจารย์ที่ปรึกษา

(สุเธียร เกียรติสุนทร)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบควบคุมแบบดิจิตอล

ฉัตรชัย วาจาเกียรติ
วิทยา รุ่งเจริญวัฒนา
สุเธียร เกียรติสุนทร อาจารย์ที่ปรึกษา
ปีการศึกษา 2534

บทคัดย่อ

โครงงานระบบควบคุมแบบดิจิตอล เป็นโครงงานที่พัฒนาต่อจากโครงงานระบบควบคุมขบวนการด้วยคอมพิวเตอร์ ในปีการศึกษา 2533 โดยมีวัตถุประสงค์เพื่อพัฒนาการใช้งานโปรแกรมเดิมในส่วนการแสดงผลที่ติดต่อกับผู้ใช้ โดยแบ่งการแสดงผลเป็นสองรูปแบบ คือ การแสดงผลชนิดตัวอักษร และการแสดงผลในลักษณะภาพเคลื่อนไหว โดยมีการทำงานเป็นสามส่วนใหญ่ ๆ คือ การเข้าถึงข้อมูลการควบคุม การสร้างภาพหลัก และการนำข้อมูลมาจัดรูปแบบเพื่อการแสดงผล

จากการทดสอบโปรแกรม การทำงานยังไม่อยู่ในระดับที่น่าพอใจ ปัญหาหลักอยู่ที่การนำภาพหลักมาประมวลผลกับค่าสัญญาณในโปรแกรมควบคุม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DIGITAL CONTROL SYSTEM

Chatchai Wajakiat

Vittaya Rungcharearnwattana

Suthian Kiatsoonthorn, Advisor

Academic Year 1991

Abstract

The Digital System Control Project is devoloped from The Computer Process Control Project ;in education year 1990 , purposed to develope the operator interface of the old . It composes of two present mode ;text mode and graphics mode, thatcan classify to three work section ;access the control data , build the static picture and manipulate data to present.

From testing, the result of program is unsatisfy. The main problem is the manipulate of the ststic picture and the signal in the control program.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

หน้า

รูปที่ 1 [illegible] 1

รูปที่ 2 แสดงโครงสร้างของระบบควบคุมแบบผสม 3

รูปที่ 3 แสดงโครงสร้างของระบบควบคุมแบบรวมศูนย์ 4

รูปที่ 4 แสดงโครงสร้างของระบบควบคุมแบบกระจาย 6

รูปที่ 5 แสดงการติดตั้งการควบคุมแบบเดี่ยว 8

รูปที่ 6 แสดงการติดตั้งการควบคุมแบบรวมศูนย์ 9

รูปที่ 7 แสดงการติดตั้งการควบคุมแบบรวมกระจาย 10

รูปที่ 8 แสดงการติดต่อในระดับสูงแบบรวมศูนย์ 14

รูปที่ 9 แสดงการติดต่อในระดับสูงแบบโมดูล 15

รูปที่ 10 แสดงวิธีการแสดงผลแบบเฉลี่ย 17

รูปที่ 11 แสดงตัวอย่างการแสดงสถานของโรงงาน 17

รูปที่ 12 แสดงตัวอย่างการแสดงพื้นที่โดยรวม 18

รูปที่ 13 แสดงตัวอย่างการแสดงผลระดับกลุ่ม 19

รูปที่ 14 แสดงตัวอย่างการแสดงแบบเทอร์เน็ต 23

รูปที่ 15 แสดงส่วนประกอบต่าง ๆ ของระบบ 26

รูปที่ 16 แสดงการดึงข้อมูลจากบล็อคข้อมูล 40

รูปที่ 17 แสดงการดึงข้อมูลจากพอร์ต 41

รูปที่ 18 แสดงภาพที่อ่านจากหน่วยความจำ 42

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้าที่
- บทนำ	1
- ส่วนติดต่อกับผู้ใช้งาน	8
- หลักการทำงาน	22
- รายละเอียดการทำงาน	30
- ผลการทดลอง	40
- สรุปผล	43
- กิตติกรรมประกาศ	45
- หนังสืออ้างอิง	46

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

หน้า

ตารางที่ 1 แสดงการเปรียบเทียบระบบควบคุมแบบต่าง ๆ 7

ตารางที่ 2 แสดงเงื่อนไขการเข้ารหัสของไฟล์สกุล pcx 36

ตารางที่ 2 แสดงโครงสร้างไฟล์สกุล pcx ใน 128ไบต์แรก 37

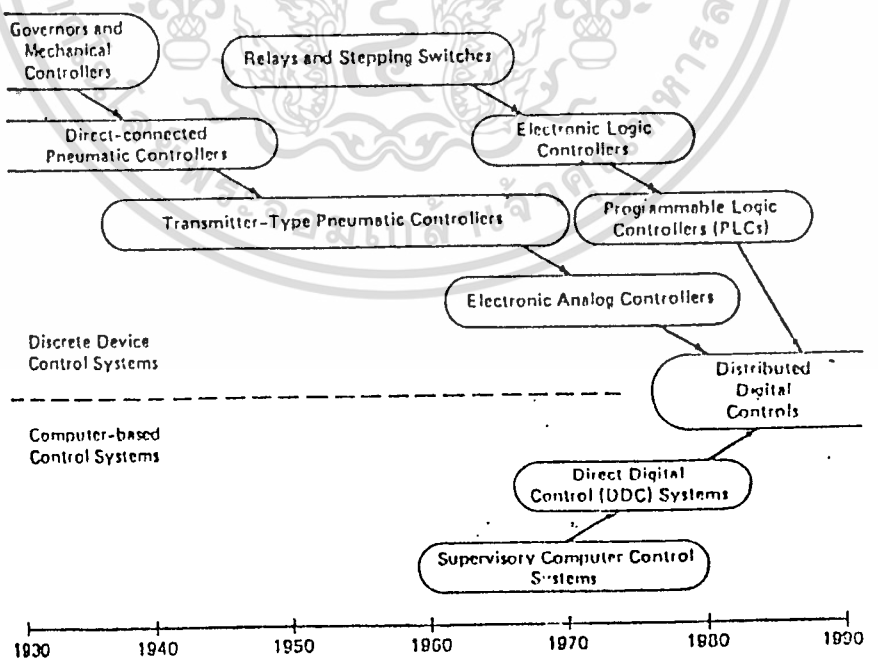
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทนำ

ประวัติศาสตร์ของความพยายามของมนุษย์ในการควบคุม ขบวนการทางอุตสาหกรรมให้เป็นแบบอัตโนมัติได้มีมานานแล้ว ตั้งแต่การสร้างเทอร์โมสตัสสำหรับควบคุมเตาเผาในปี ค.ศ.1620 แต่ระบบควบคุมเริ่มเป็นรูปร่างอย่างจริงจังเมื่อประมาณห้าสิบปีที่ผ่านมานี้เอง

การที่ระบบควบคุมเริ่มมีการพัฒนาอย่างจริงจังตั้งแต่ปี ค.ศ. 1930 จนถึงปัจจุบันนั้นมีตัวกระตุ้นอยู่สองประการ คือ ความต้องการของผู้บริโภค และ ความก้าวหน้าทางเทคโนโลยีที่ทำให้ระบบควบคุมมีประสิทธิภาพมากขึ้น วิวัฒนาการของการพัฒนาระบบควบคุมทางอุตสาหกรรมนั้นสามารถดูได้จากรูปประกอบที่ 1 ซึ่งจากภาพจะเห็นได้ว่าระบบควบคุมที่อยู่ในยุคปัจจุบันนี้คือระบบการควบคุมโดยใช้ดิจิตอลแบบกระจาย (Distributed Digital Controls)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 1 แสดงวิวัฒนาการของระบบควบคุม

การพัฒนาระบบควบคุมโดยใช้คอมพิวเตอร์

(Computer-based Control System Developments)

หลักการของระบบควบคุมโดยใช้คอมพิวเตอร์ไม่ได้เป็นของใหม่เท่าใดนัก ตามจริงแล้วก็คือการนำข้อมูลจากอุปกรณ์ควบคุมและอุปกรณ์วัดต่างๆในกระบวนการทำงาน (process) มาประมวลผล แล้วส่งสัญญาณควบคุมออกไป โดยเปลี่ยนการควบคุมโดยใช้มนุษย์เป็นการทำงานโดยอัตโนมัติ และเปลี่ยนการทำงานแบบอนาลอคมาเป็นแบบดิจิตอล ตัวอย่างที่เห็นได้ชัดเจนก็คือ เครื่องโปรแกรมเมเบอร์ลอจิกคอนโทรลเลอร์ (Programmable Logic Controller) ซึ่งมีการนำมาใช้งานอย่างชัดเจนตั้งแต่ยุคปี ค.ศ. 1970 ระบบควบคุมโดยใช้คอมพิวเตอร์เริ่มมีความจำเป็นต่อการใช้งานมากขึ้นเรื่อย ๆ เนื่องจากระบบควบคุมยุคใหม่มีความสลับซับซ้อนมากขึ้นและต้องการความแม่นยำในการควบคุมมากขึ้น งานแรกที่ระบบควบคุมนำคอมพิวเตอร์มาใช้งานได้แก่งานการแสดงผลและดูแลการควบคุม (Supervisory Control) หลังจากนั้นได้มีการนำคอมพิวเตอร์มาควบคุมการปรับจุดเป้าหมาย (setpoint) เพื่อเพิ่มประสิทธิภาพการในการควบคุม ซึ่งสัญญาณอ้างอิงเหล่านี้จะถูกแปลงแล้วส่งไปยังตัวควบคุมชนิดอนาลอค และในขั้นสุดท้ายคือการนำคอมพิวเตอร์มาใช้เป็นตัวควบคุมโดยตรงเลยซึ่งเรารู้จักกันในชื่อว่า direct digital control(DDC) ซึ่งระบบ DDC นี้ได้เริ่มนำมาใช้งานครั้งแรกในโรงงานปิโตรเคมี เมื่อปี ค.ศ. 1963 แต่ในช่วงแรกของการใช้งานนั้นเพื่อความปลอดภัย จึงมีการนำระบบควบคุมแบบอนาลอคมาใช้ร่วมด้วยเพื่อสำรองในกรณีระบบคอมพิวเตอร์เกิดชัดข้อง ข้อดีของระบบควบคุมโดยใช้คอมพิวเตอร์นี้มีหลายประการ เช่น ในการปรับค่าพารามิเตอร์ต่างๆ จะไม่เกิดการดริฟท์(drift) ความสามารถในการสร้างการควบคุมที่สลับซับซ้อน และความสามารถในการปรับค่าพารามิเตอร์ตามสภาวะที่เปลี่ยนแปลงได้อย่างเหมาะสม (optimum control)

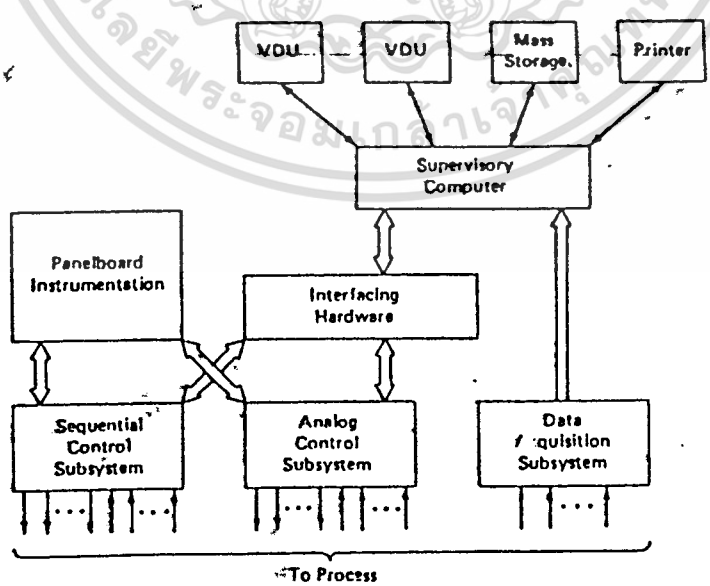
โครงสร้างของระบบควบคุมโดยใช้คอมพิวเตอร์

จากการพัฒนาระบบควบคุมมาจนถึงปัจจุบันเราสามารถแบ่งโครงสร้างระบบควบคุมได้เป็น 3 ลักษณะ คือ

Hybrid Architecture

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ในการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ในการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น ไม่อนุญาตให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โครงสร้างแบบผสม(hybrid system) เป็นผลรวมของการใช้งาน ของฮาร์ดแวร์การควบคุมของแบบดิสครีต (discretecontrol hardware) และ ฮาร์ดแวร์ของคอมพิวเตอร์(computer hardware) ที่ตำแหน่งศูนย์กลางของการควบ คุมเมื่อใช้งานเกี่ยวกับกับฟังค์ชันที่เพิ่มขึ้นมาตามความต้องการในโครงสร้างแบบนี้ การควบคุมแบบท้องถิ่น(local control) ในหน่วยของขบวนการถูกเพิ่มเติมโดยใช้ สัญญาณอนาลอกแบบดิสครีต(discreter analog) และตัวควบคุมแบบซีเควนเชียล (sequentialdiscrete analog) หรือ PLC การวัดของแผงควบคุมติดต่อกับตัวควบคุมเหล่านี้ โดยใช้สำหรับผู้ควบคุมและติดตั้งอยู่ที่ห้องควบคุมกลาง (central control room area) นอกจากนี้ยังมีการนำเอาซุปเปอร์ไวเซอร์คอมพิวเตอร์ (supervisor computer) และระบบการรวบรวมข้อมูลมาช่วยงานในการจัดการต่างๆ ของขบวนการ รวมไปถึงการควบคุมขบวนการให้เหมาะสม การเตือนภัย การส่งข้อมูลการเก็บข้อมูล และการนำเอาข้อมูลเหล่านี้มาใช้งาน ส่วนของการติดต่อกับผู้ควบคุมจะประกอบด้วยอุปกรณ์แสดงผลตั้งแต่ 1 ตัวขึ้นไป

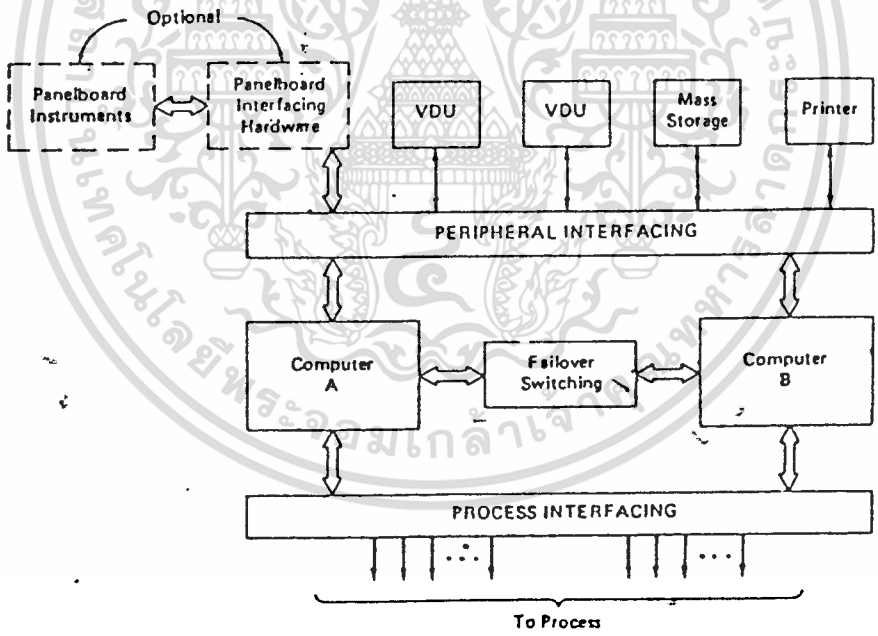


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2 แสดงโครงสร้างระบบควบคุมแบบผสม

2. Central Computer System

ส่วนโครงสร้างชนิดนี้สามารถแสดงให้เห็นได้ในรูป 1.3 ซึ่งเป็นระบบที่เพิ่มเติมอุปกรณ์ที่มีประสิทธิภาพสูงเข้าไปในห้องควบคุม โดยทั่วไปจะต้องมีการนำเอาคอมพิวเตอร์สำรอง (redundancy) มาเพิ่มเข้าไปในกรณีที่คอมพิวเตอร์หลักไม่สามารถทำงานได้ตามปรกติส่วนการติดต่อใช้งานของผู้ควบคุมสามารถทำได้โดยผ่านอุปกรณ์รับและแสดงผลเหมือนกับโครงสร้างการควบคุมแบบผสมที่กล่าวมาแล้วข้างต้น ส่วนทางเลือกอื่นๆ ได้แก่ การใช้งานคอมพิวเตอร์ร่วมกับแผงวัดของการควบคุม (panel instrumentation) ทำให้ผู้ควบคุมทำการติดต่อในระดับแรกได้เหมือนตามปกติ



รูปที่ 3 แสดงโครงสร้างระบบควบคุมแบบรวมศูนย์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และ 3. Distributed Control System

ระบบควบคุมแบบกระจาย (Distributed Control System)

จากรูปที่ 4 ซึ่งแสดงโครงสร้างของระบบควบคุมแบบกระจายไว้ จะเห็นได้ว่าโครงสร้างประกอบด้วยส่วนประกอบที่มากกว่าโครงสร้างการควบคุมแบบอื่น ๆ ซึ่งอธิบายโดยสรุปได้ดังนี้

- Local Control Unit (LCU) : เป็นหน่วยงานที่เล็กที่สุดในระบบซึ่งมีความสามารถในการควบคุมแบบปิด (closed-loop control)ได้ ตัว LCU จะเป็นตัวรับและส่งสัญญาณเข้ากับกระบวนการผลิต (process) โดยตรง

- Low-level Human Interface (LLHI) : เป็นอุปกรณ์ให้ผู้ควบคุมสามารถติดต่อกับ LCU ในการปรับเปลี่ยนค่าพารามิเตอร์ต่าง ๆได้โดยตรง

- Data Input/Output Unit (DI/OU) : เป็นอุปกรณ์ในการรับส่งสัญญาณจากกระบวนการผลิตเพื่อนำไปใช้ในงานอื่น ๆ ต่อไป

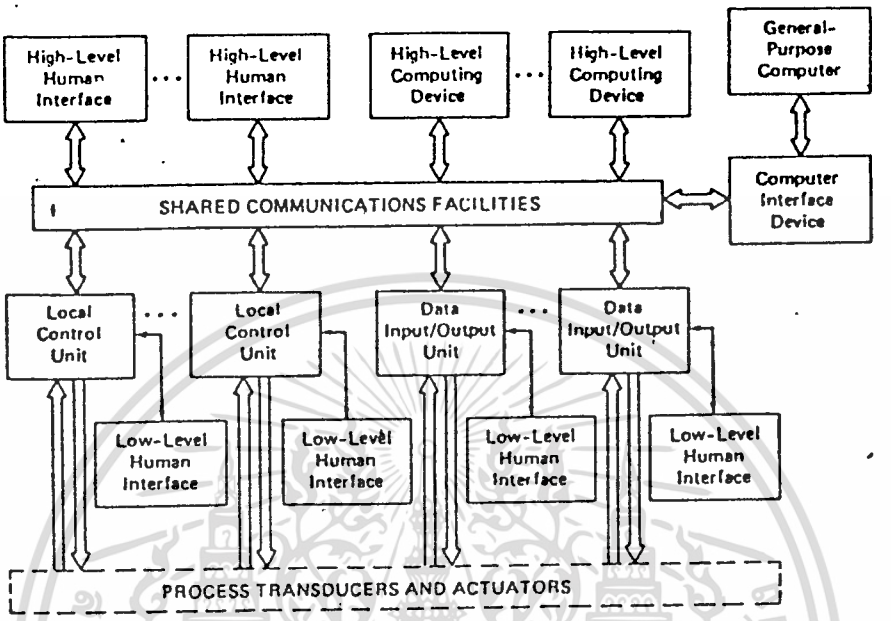
- High-Level Human Interface (HLHI) : เป็นชุดอุปกรณ์ที่ทำหน้าที่คล้ายกับ LLHI แต่ได้เพิ่มประสิทธิภาพการทำงาน และความสะดวกในการใช้งานให้มากขึ้น นอกจากนี้ยังสามารถติดต่อกับอุปกรณ์ควบคุมตัวอื่น โดยผ่านระบบสื่อสารข้อมูลร่วมได้

- High-Level Computering Device (HLCD) : เป็นชุดคอมพิวเตอร์ที่ทำหน้าที่ในการจัดการ เกี่ยวกับกระบวนการผลิตทั้งหมดในรูปของการแสดงผล การรายงานความผิดพลาดและการปรับค่าพารามิเตอร์ต่าง ๆ ของทั้งระบบโดยผ่านระบบสื่อสารข้อมูลร่วม

- Computer Interface Device (CID) : เป็นอุปกรณ์สำหรับคอมพิวเตอร์นอกระบบที่จะเข้ามาติดต่อกับระบบควบคุมนี้

- ระบบสื่อสารร่วม : เป็นอุปกรณ์ทั้งฮาร์ดแวร์และซอฟท์แวร์ที่ทำหน้าที่ในการจัดสรรข้อมูลระหว่างหน่วยการทำงานต่างๆ ภายในระบบให้ได้รับข้อมูลที่หน่วยการทำงานนั้นเกี่ยวข้องไปปฏิบัติงาน

เอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4 แสดงโครงสร้างระบบควบคุมแบบกระจาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FEATURE	HYBRID ARCHITECTURE	CENTRAL COMPUTER ARCHITECTURE	DISTRIBUTED ARCHITECTURE
1. Scalability and expandability	Good due to modularity	Poor—very limited range of system size	Good due to modularity
2. Control capability	Limited by analog and sequential control hardware	Full digital control capability	Full digital control capability
3. Operator interfacing capability	Limited by panelboard instrumentation	Digital hardware provides significant improvement for large systems	Digital hardware provides improvement for full range of system sizes
4. Integration of system functions	Poor due to variety of products	All functions performed by central computer	Functions integrated in a family of products
5. Significance of single-point failure	Low due to modularity	High	Low due to modularity
6. Installation costs	High due to discrete wiring and large volume of equipment	Medium—saves control room and equipment room space but uses discrete wiring	Low—savings in both wiring costs and equipment space
7. Maintainability	Poor—many module types, few diagnostics	Medium—requires highly trained computer maintenance personnel	Excellent—automatic diagnostics and module replacement

ส่วนติดต่อกับผู้ใช้งาน
OPERATOR INTERFACE

ในระบบการควบคุมอัตโนมัติโดยทั่วไปจะมีบุคคลที่เกี่ยวข้องอยู่หลายฝ่าย เราสามารถจำแนกได้เป็น 2 กลุ่มใหญ่ๆ ดังนี้

1.Instrumentation and Control System Engineer เป็นบุคคลที่มีความรู้ความสามารถเกี่ยวกับการติดตั้งอุปกรณ์ที่ใช้ในระบบ การปรับปรุงค่าพารามิเตอร์ต่างๆ ของตัวควบคุม

2.Plant Operator เป็นผู้ดูแลควบคุมชบวนการให้ไปถึงจุดที่ต้องการ

ผู้ควบคุมสามารถติดต่อกับการควบคุมชบวนการแบบกระจาย (Distributed Control System) นั้นสามารถทำได้ 2 ระดับ คือ

1.การติดต่อกับบุคคลในระดับต่ำ(LLHI) เป็นการติดต่อโดยตรงระหว่างผู้ควบคุมกับหน่วยควบคุม(Local Control Unit) หรืออุปกรณ์อินพุทเอาท์พุทอื่นๆ

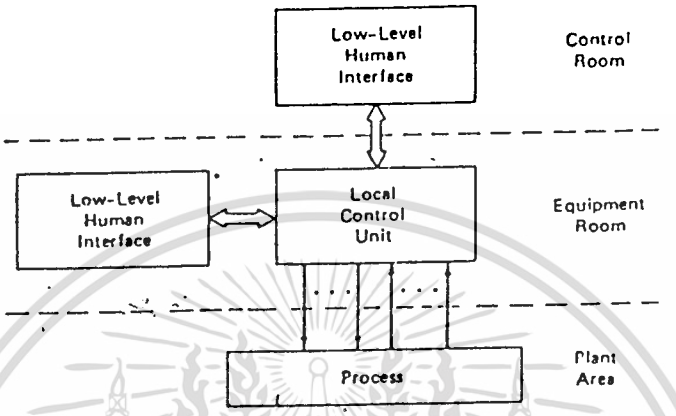
2.การติดต่อกับบุคคลในระดับสูง(HLHI) เป็นการติดต่อระหว่างผู้ควบคุมกับหน่วยควบคุมหรืออุปกรณ์อินพุทเอาท์พุทโดยผ่านระบบสื่อสาร

ระบบการติดต่อกับบุคคลในระดับต่ำเป็นการติดต่อผ่านแผงควบคุมการวัด (Panelboard Instrument) เป็นการติดต่อระหว่างผู้ควบคุมกับหน่วยควบคุมในระยะใกล้ๆ

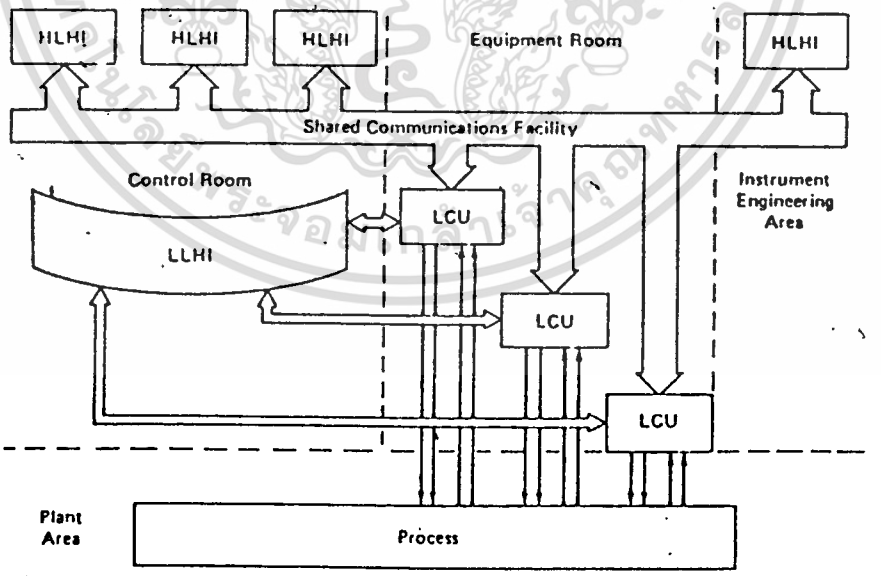
ส่วนการติดต่อในระดับสูงมีอุปกรณ์ต่างๆ มากมายช่วยสนับสนุน เช่น จอภาพ เครื่องพิมพ์ เป็นต้น สามารถทำการติดต่อ ณ ตำแหน่งใดก็ได้ โดยผ่านเครือข่ายสื่อสาร

รูป 5 เป็นลักษณะการติดตั้งที่ง่ายและเล็กมีหน่วยควบคุมเพียงตัวเดียวติดตั้งอยู่ในห้องอุปกรณ์(plant equipment room) การติดต่อในระดับต่ำสามารถทำได้ที่ห้องอุปกรณ์หรือห้องควบคุม(plant control room) ในรูปนี้เหมาะสมกับอุปกรณ์ที่มีขนาดเล็ก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5 การติดตั้งการควบคุมแบบเดี่ยว(Stand-Alone Control)



รูป 6 การติดตั้งการควบคุมแบบรวมศูนย์

(Geographically Centralized Control)

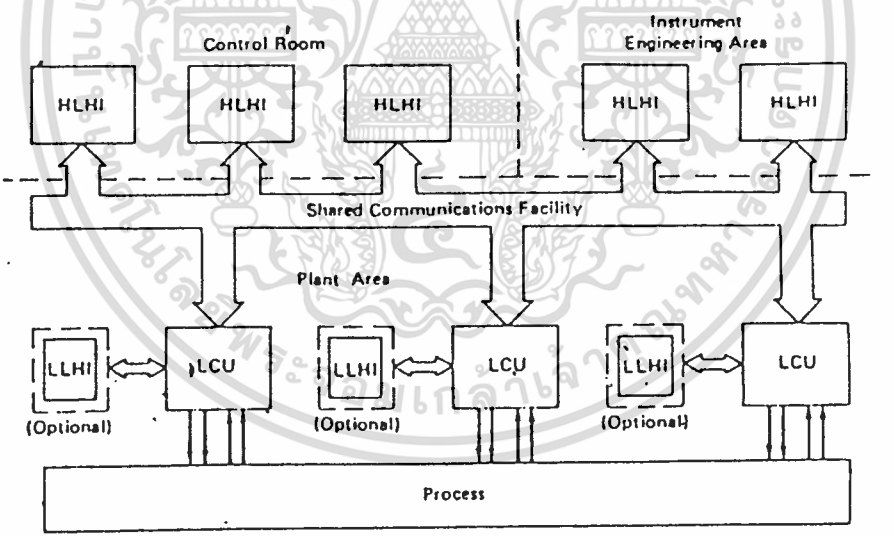
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูป 6 เป็นรูปแบบที่นิยมใช้กันมากมีหน่วยควบคุมหลายตัวรวมอยู่ในห้องอุปกรณ์

007713

การติดตั้งในระดับต่ำและระดับสูงสามารถกระทำได้ที่ห้องควบคุม การติดตั้งในระดับสูงเป็นการใช้ฟังก์ชั่นต่างๆ ของการควบคุมขบวนการหรือสำหรับให้วิศวกรวัดคุมทางอุตสาหกรรมทำการติดต่อกับกระบวนการ ส่วนการติดต่อในระดับใช้ในกรณีที่การติดต่อในระดับสูงไม่สามารถใช้งานได้

รูปที่7 เป็นการควบคุมแบบกระจายที่สมบูรณ์แบบที่สุด โดยหน่วยควบคุมแต่ละตัวจะกระจายกันอยู่ตามกระบวนการต่างๆ ในห้องควบคุมและห้องของวิศวกรวัดคุมฯ จะใช้การติดต่อในระดับสูงผ่านกระบวนการเพื่อทำการควบคุมและใช้ฟังก์ชั่นต่างๆ ส่วนการติดต่อในระดับล่างจะเป็นการติดต่อกับกระบวนการในกรณีที่เกิดปัญหากับการติดต่อในระดับสูงหรือเพื่อการบำรุงรักษา



รูปที่ 7 การติดตั้งการควบคุมแบบกระจาย

(Geographically Distributed Control System)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

OPERATOR INTERFACE REQUIREMENT

1. PROCESS MONITORING องค์ประกอบพื้นฐานที่ต้องมี

1. สถานะต่างของกระบวนการแต่ละอัน สามารถแสดงออกมาให้เห็นได้

2.การเรียกชื่อตัวแปรกระบวนการ จะเรียกในรูปของ ป้ายชื่อ(Tag Name) หรือชื่อที่ตั้งโดยวิศวกรวัดคุมทางอุตสาหกรรม

3. หน่วยของตัวแปรกระบวนการต้องอยู่ในหน่วยของวิศวกร

ส่วนหน้าที่อื่นๆ เช่น การตรวจสอบความผิดปกติของสถานการณ์ต่างๆ ของกระบวนการและรายงานให้ผู้ควบคุมทราบ องค์ประกอบมีดังนี้

1.ต้องแสดงสถานะต่างๆ ของกระบวนการให้เป็นที่เข้าใจได้ง่าย เช่น ระดับสูงต่ำอาจแสดงด้วยสีต่างกัน

2.การรายงานผลทำเหมือนกับการแสดงสถานะของกระบวนการ

3.ถ้าระบบพบว่ามีสัญญาณเตือนเข้ามาหลายอันผู้ควบคุมจะต้องสามารถเข้าถึงสัญญาณเตือนเหล่านั้นได้ทั้งหมด

คุณสมบัติในส่วนแสดงผลนั้นไม่เพียงแต่จะบอกสถานะต่างๆ ของกระบวนการเท่านั้น แต่ยังต้องสามารถบอกแนวโน้มได้ว่าขณะนี้กระบวนการจะมีแนวไปทางไหน จึงจะทำให้ผู้ควบคุมสามารถควบคุมกระบวนการได้อย่างถูกต้อง ในลักษณะเช่นนี้เราเรียกว่าการแสดงแนวโน้มมีคุณสมบัติดังนี้

1.สามารถแสดงเป็นกลุ่มของตัวแปรต่างๆ ที่มีความเกี่ยวข้องกัน และแสดงในหน่วยพื้นฐานทางวิศวกรรม

2.ผู้ควบคุมสามารถอ่านค่าของตัวแปรต่างๆ ทั้งในปัจจุบันและในอดีต

3.ในรูปกราฟเดียวกัน ต้องมีข้อความบางอย่างเพื่อที่จะให้ความสะดวกแก่การเข้าใจ

2.Process Control

ฟังก์ชั่นพื้นฐานมีดังนี้

1.ผู้ควบคุมสามารถเข้าถึงการควบคุมทั้งในลักษณะต่อเนื่องและในลักษณะเป็นลำดับอย่างรวดเร็ว

2.ในการควบคุมอย่างต่อเนื่องผู้ใช้สามารถเปลี่ยนพารามิเตอร์ได้อย่างรวดเร็วและง่าย เช่น การเปลี่ยนโหมดของการควบคุม การเปลี่ยนค่าอ้างอิง เป็นต้น

3.ผู้ควบคุมสามารถเข้าถึงการควบคุมแบบลอจิกได้ เช่น การปิดเปิดปั๊ม การปิดเปิดวาล์ว และสามารถแสดงสถานะต่างๆ ให้ผู้ควบคุมได้เห็น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น และห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเอกสารทุกครั้งที่มีการนำไปใช้

3.Process Diagnostics

การเตือนสัญญาณและการควบคุมกระบวนการภายใต้สถานการณ์ที่ปกติจะใช้ฟังก์ชั่นที่ธรรมดาเมื่อเทียบกับการทำงานในสถานะที่ผิดปกติที่เกิดจากอุปกรณ์ในระบบทำงานผิดปกติหรือปัญหาจากการวัดที่ผิดพลาด ในการติดต่อกับผู้ควบคุมต้องมีข้อมูลเพียงพอเพื่อให้ผู้ควบคุมสามารถแก้ปัญหาได้อย่างรวดเร็ว

ขั้นตอนในการทำงานลำดับแรกคือ ถ้าปัญหาเกิดจากอุปกรณ์ควรใช้การวินิจฉัยตามขั้นตอนดังนี้

1.ดำเนินการทดสอบต่อไปกับการตรวจสอบอุปกรณ์แปลงสัญญาณที่เกี่ยวข้อง

2.ดำเนินการทดสอบต่อไปเกี่ยวกับการควบคุม ได้แก่ ตัวควบคุม อุปกรณ์สื่อสาร อุปกรณ์คำนวณและอุปกรณ์ที่ติดต่อกับผู้ควบคุม

สรุปขั้นตอนการวินิจฉัยของการควบคุมแบบอัตโนมัติ เมื่อตรวจสอบพบความผิดพลาดเกิดขึ้น

1.การเตือนภัยลำดับแรกจะบอกผู้ควบคุมถึงอันตรายที่จะเกิดขึ้นก่อน

2.ลำดับความสำคัญของการเตือนภัย

3.การวินิจฉัยที่ซับซ้อนเพิ่มมากขึ้น เกิดจากการรวมกันของข้อมูลเกี่ยวกับการเตือนภัยและข้อมูลของกระบวนการที่เกิดจากการทำงานที่ผิดพลาดของอุปกรณ์

4.Process Record Keeping

ในการใช้มนุษย์เก็บรวบรวมข้อมูลของตัวแปรต่างๆ ในกระบวนการเป็นเรื่องที่น่าเบื่อ ปัญหานี้จะหมดไปเมื่อนำเอาคอมพิวเตอร์มาใช้ช่วยในการจัดพิมพ์รายงานโดยมีคุณสมบัติดังนี้

1.สามารถรายงานผลที่เกิดจากการป้อนของข้อมูลของผู้ควบคุม

2.แสดงสถานะของสัญญาณเตือนภัยต่างๆ ที่เกิดขึ้นโดยอยู่ในรูปที่ผู้ควบคุมสามารถเข้าใจได้

3.สามารถปรับช่วงเวลารายงานผลของอุปกรณ์ควบคุมแต่ละตัวได้อย่างอิสระ เช่น ตัวแปรบางตัวมีการเปลี่ยนแปลงที่เร็วมากจำเป็นต้องมีการรายงานผลที่ถี่ ส่วนตัวแปรบางอย่างมีผลตอบสนองที่ช้าก็ตั้งเวลาให้นานขึ้น

4.สามารถเก็บรวบรวมข้อมูลต่างๆ เป็นจำนวนมากและนาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การติดต่อในระดับต่ำ(LOW LEVEL OPERATOR INTERFACE)

การติดต่อในระดับล่างในการควบคุมแบบกระจายนั้นเป็นการติดต่อกัน

โดยตรงระหว่างหน่วยควบคุมกับผู้ควบคุม

ข้อดีของการติดต่อในระดับล่าง

1. ราคาถูกในกรณีที่มีจำนวนลูปควบคุมจำนวนน้อย

2. การเรียนรู้การใช้งานสามารถทำได้อย่างรวดเร็ว

3. ใช้เป็นตัวเสริมในกรณีที่การติดต่อในระดับสูงไม่สามารถทำงานได้

อุปกรณ์ที่ใช้งานร่วมกับการติดต่อในระดับต่ำ เช่น ตัวควบคุม ตัวแสดงสถานะ รายงานผลแนวโน้มและแสดงความผิดปกติที่เกิดขึ้น

<u>การนำเอาการติดต่อในระดับต่ำไปประยุกต์ใช้งาน</u>

การติดตั้งในช่วงแรกผู้ใช้และผู้ขายควรห่วงใยในข้อที่บุคลากรด้านการปฏิบัติการได้รับเทคโนโลยีใหม่ๆ ทั้งนี้ต้องเกี่ยวข้องใน 2 แง่มุมดังนี้

1. ใช้ไมโครโปรเซสเซอร์ช่วยในการควบคุมแบบปิด

2. ใช้อุปกรณ์การติดต่อกับมนุษย์แบบใหม่อย่างเช่น จอภาพ ในเครื่องมือพื้นฐาน ดังนั้นความรับผิดชอบในการติดตั้งได้นำวิธี "belt and suspender" มาออกแบบอุปกรณ์การติดต่อโดยใช้กับแผงเครื่องมือวัดและส่วนควบคุมจอภาพ

การติดต่อในระดับสูง (HIGH LEVEL OPERATOR INTERFACE)

การติดต่อในระดับสูงเป็นระบบควบคุมที่ผู้ควบคุมสามารถติดต่อกับหน่วยควบคุมหลายๆ ตัว โดยอาศัยเครือข่ายสื่อสาร

ข้อดีของการติดต่อในระดับสูง

1. พื้นที่ของห้องควบคุมลดลง

2. สามารถออกแบบการติดต่อกับผู้ปฏิบัติการในกระบวนการเฉพาะเจาะจงที่ยืดหยุ่นได้

3. ไมโครโปรเซสเซอร์มีราคาถูกเหมาะสำหรับใช้งาน

<u>โครงสร้างทางสถาปัตยกรรม</u>

อุปกรณ์ที่ใช้เป็นส่วนประกอบในการควบคุมแบบกระจาย เช่น อุปกรณ์แสดงผล คีย์บอร์ดหรืออุปกรณ์ทางอินพุท ไมโครโปรเซสเซอร์และหน่วยความจำ อุปกรณ์ที่ใช้ในการสื่อสาร เป็นต้น อย่างไรก็ตาม โครงสร้างทางสถาปัตยกรรมของการติดต่อในระดับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น และห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารนี้ทุกครั้งที่มีการนำไปใช้

สูง ยังขึ้นอยู่ความต้องการของผู้ใช้และตลาดด้วย

รูป 8 จะพบว่ามีหน่วยประมวลผลกลางทำหน้าที่ในการคำนวณ การจัดการจัดเก็บข้อมูล การโอนย้ายข้อมูล การติดต่อกับอุปกรณ์แสดงผลรวมทั้งอุปกรณ์สื่อสาร

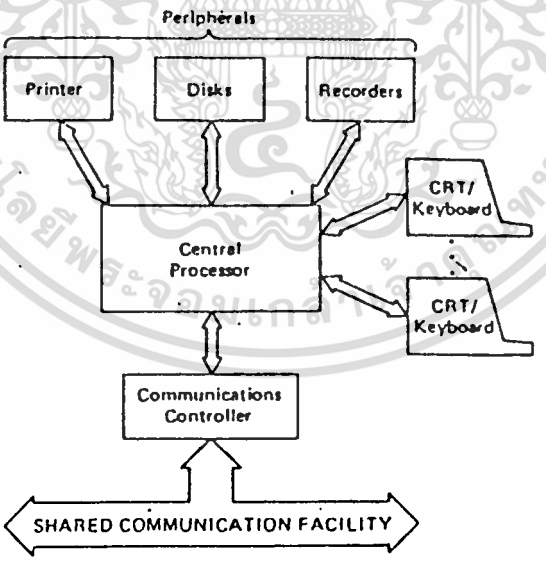
ส่วนดีของลักษณะโครงสร้างข้างบน คือ

1. ข้อมูลต่างๆ ของกระบวนการจะถูกรวบรวมไว้ที่เดียวกันโดยอาศัยเครือข่ายสื่อสารและนำออกแสดงโดยผ่านทางจอภาพ

2. มีการแบ่งการใช้งานอุปกรณ์ให้เหมาะสมกับการใช้งานในขณะนั้น

สำหรับข้อเสียของการควบคุมแบบนี้ คือ

1. หน่วยประมวลผลกลางและหน่วยประมวลผลเพียงอันเดียวเป็นข้อจำกัดของการต่อจำนวนลูปของการควบคุมและจำนวนข้อมูลที่รับเข้ามาได้ ซึ่งจะทำให้กินเวลานาน



รูปที่ 8 การติดตั้งของการติดต่อกับบุคคลในระดับสูงแบบรวมศูนย์

(Centralized HLOI Configuration)

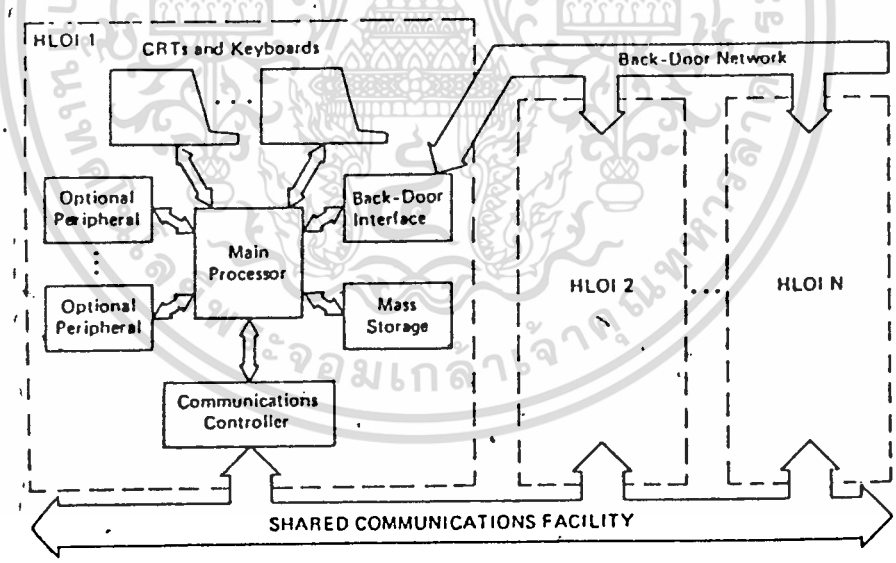
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้ ... ใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. ขนาดไม่เหมาะสมกับราคา

จากข้อจำกัดข้างต้นทำให้การควบคุมแบบกระจายโดยทั่วไปจะนิยมใช้การกระจายการติดต่อระดับสูง โดยในลักษณะนี้หน่วยการติดต่อในระดับสูงหลายๆ หน่วย โดยในแต่ละหน่วยเรียกว่าโนด หรือการแบ่งเครือข่ายการติดต่อสื่อสารระหว่างกัน ในแต่ละหน่วยจะมีอุปกรณ์แสดงผล คีย์บอร์ด และอุปกรณ์ภายนอกอื่นๆ เป็นของส่วนตัว

รูป 9 แสดงลักษณะการกระจายการติดต่อระดับสูงที่ใหม่ที่สุดที่สะดวกต่อการใช้งานในอนาคตหากจำเป็นต้องขยายลูปของการควบคุมมากขึ้น ส่วนประกอบมีดังนี้

1. มีหน่วยแสดงผลจำนวนมากกระจายอยู่ทั่วทั้งกระบวนการ
2. การเปลี่ยนแปลงข้อมูลทำได้โดยผ่านคีย์บอร์ด
3. มีหน่วยความจำขนาดใหญ่ที่สามารถเก็บข้อมูลเป็นจำนวนมาก
4. สามารถติดต่อกับอุปกรณ์ที่แสดงผลในรูปแนวโน้ม สัญญาณเสียงและอุปกรณ์ฮาร์ดแวร์ภายนอกอื่นๆ



รูปที่ 9 การติดตั้งของการติดต่อระดับสูงแบบโมดูล (Modular HLOI)

<u>ส่วนประกอบอื่นๆ</u>

ไมโครโปรเซสเซอร์และหน่วยความจำ ในส่วนของการติดต่อในระดับสูงไมโครโปรเซสเซอร์ที่ใช้ต้องมีความเร็วมากกว่าไมโครโปรเซสเซอร์ที่ใช้ในหน่วยอื่นของการควบคุมแบบกระจาย คุณสมบัติของไมโครโปรเซสเซอร์ที่ใช้มีดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ในการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารนี้ทุกครั้งที่มีการนำไปใช้

1.มีขนาด 16 หรือ 32 บิต จากหลายๆ แหล่ง

2.โปรเซสเซอร์ที่ใช้ต้องอยู่ในตระกูลที่ได้มาตรฐาน

3.การทำงานเป็นแบบมัลติโปรเซสเซอร์

ส่วนหน่วยความจำที่ใช้เป็นแบบเดียวกับการติดต่อในระดับต่ำ เพียงแต่มีจำนวนมากและความเร็วสูง

-อุปกรณ์ที่ใช้สำหรับการติดต่อกับผู้ควบคุมแบ่งได้ 2 อย่าง

1.อุปกรณ์ที่ใช้รับข้อมูลเข้ามาได้แก่ คีย์บอร์ด ไลท์เพน(light pen) อุปกรณ์สำหรับการเลื่อนเคอร์เซอร์

2.อุปกรณ์ที่ใช้แสดงผลของข้อมูลได้แก่ จอภาพ อุปกรณ์ที่ส่งเสียงได้

-อุปกรณ์อื่นเพื่อให้การติดต่อการควบคุมแบบกระจายมีประสิทธิภาพมากขึ้น เช่น ฟลอปปี้ดิสก์ เครื่องพิมพ์ เครื่องพล๊อต

<u>การแสดงผลการทำงาน(Operator Display)</u>

การออกแบบโครงสร้างการแสดงผลที่เหมาะสมจะทำให้ผู้ควบคุมสามารถเข้าถึงกระบวนการได้ทั่งถึงและรวดเร็ว

วิธีการแสดงผลแบบเฉพาะ(Typical Display Hierarchy) สามารถแบ่งได้เป็นระดับต่างๆ ดังรูป 10 ดังนี้

1.Plant level เป็นการแสดงผลข้อมูลที่เกี่ยวกับกระบวนการทั้งหมด ถ้ากระบวนการใหญ่มากอาจแบ่งการแสดงผลออกเป็นหลายๆ จุด

2.Area level เป็นการแสดงผลที่เกี่ยวกับกระบวนการเพียงบางส่วน

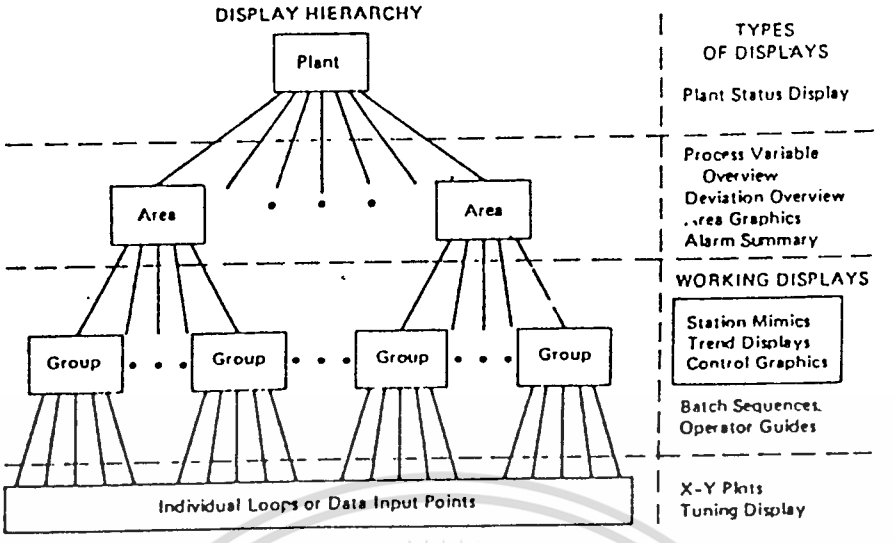
3.Group level เป็นการแสดงผลที่เกี่ยวกับลูปควบคุมและข้อมูลที่เกี่ยวข้องกับกระบวนการเดี่ยวๆ ภายในพื้นที่หนึ่งๆ

4.Loop level เป็นระดับการแสดงผลที่เกี่ยวกับลูปควบคุม

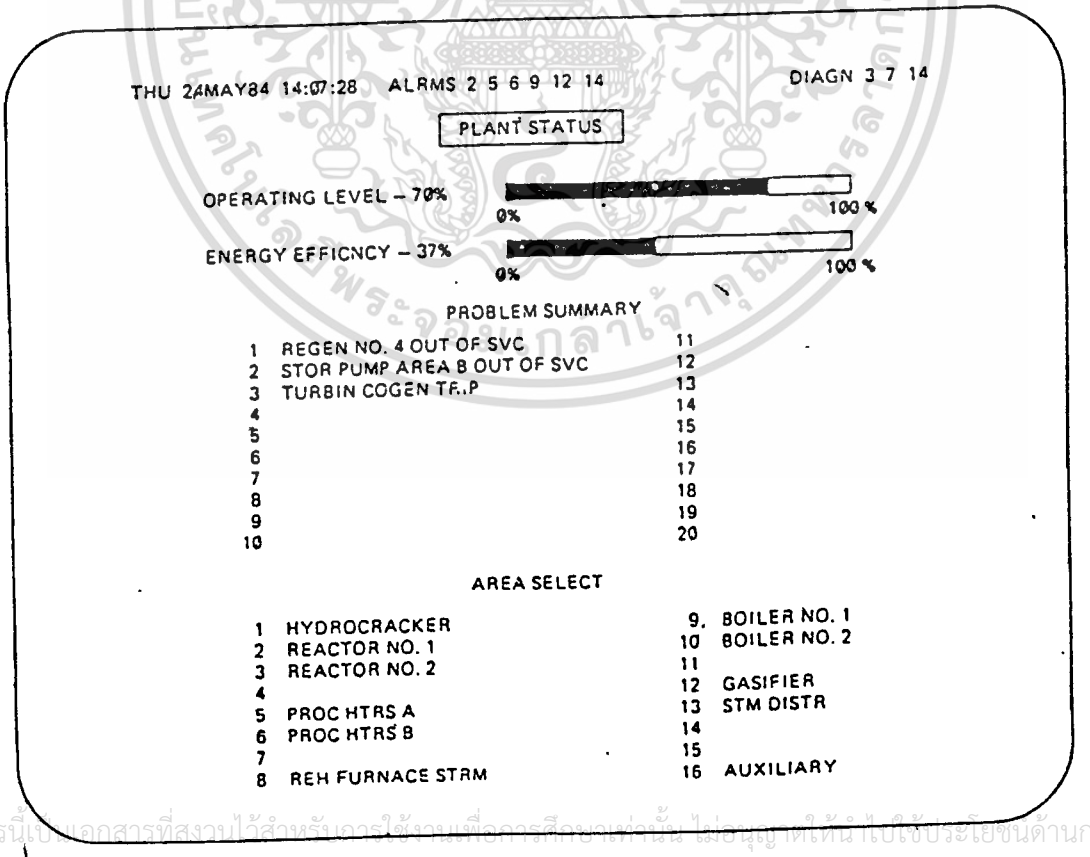
-Plant-level Display เป็นระดับการแสดงผลที่อยู่ระดับบนสุดดังรูป 11 ลักษณะการแสดงผลจะเป็นการสรุปข้อมูลต่างๆ ที่จำเป็นสำหรับผู้ควบคุมเพื่อให้การควบคุมมีประสิทธิภาพมากขึ้นและแสดงถึงจุดที่ประสบปัญหาให้ผู้ควบคุมได้รับทราบ

- Area-level Display เมื่อผู้ควบคุมได้ทราบลักษณะทั่วไปของกระบวนการผู้ควบคุมสามารถดูรายละเอียดข้างในได้อีก โดยลงมาดูใน plant area

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารนี้ทุกครั้งที่มีการนำไปใช้

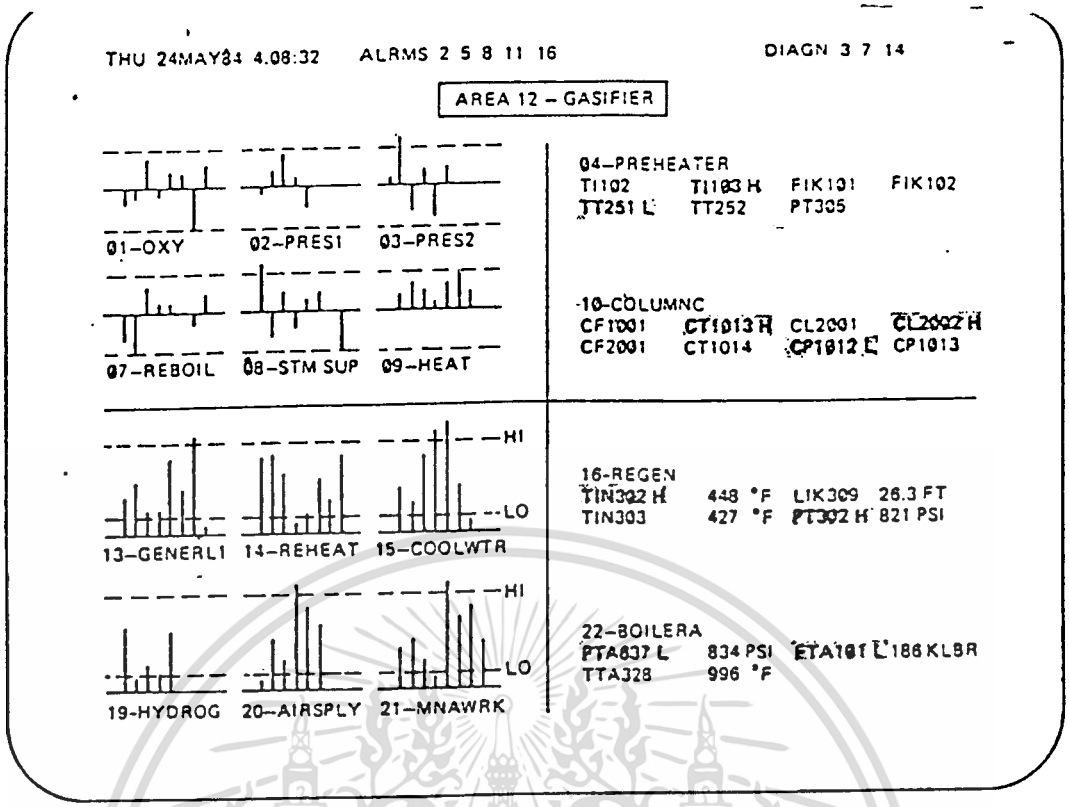


รูปที่ 10 วิธีการแสดงผลแบบเฉพาะ (Typical Display Hierarchy)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 11 ตัวอย่างการแสดงสถานะของโรงงาน



รูปที่ 12 ตัวอย่างการแสดงพื้นที่โดยรวม

การแสดงผลในระดับนี้สามารถแบ่งได้ 2 ประเภท คือ

1.Area graphic display เป็นการแสดงผลที่คล้ายกับ piping and instrument display(P&ID) หรือ mimic panel

2.Alarm summary display เป็นการแสดงผลที่แสดงสถานะเตือนภัยที่เกิดขึ้นในพื้นที่สำคัญๆ

Group-level Display

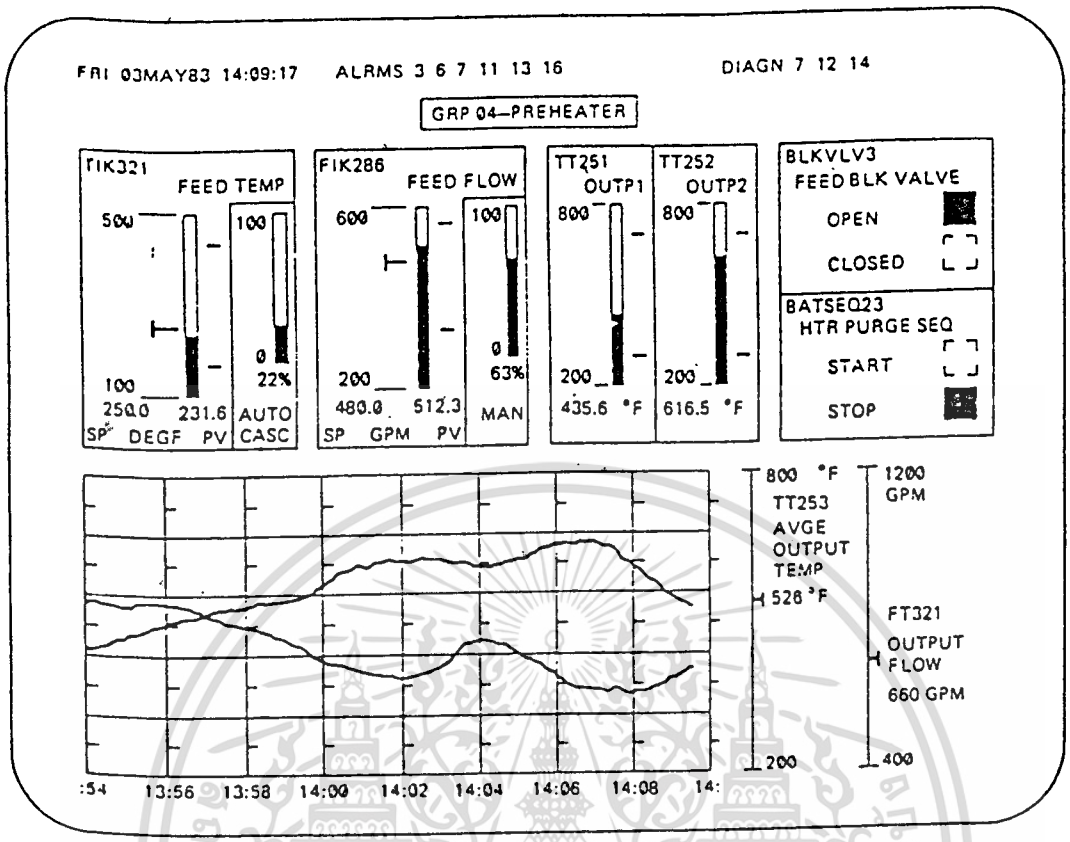
ชนิดของการแสดงผลระดับกลุ่ม(Group-level)

1.Batch control display เป็นลักษณะเมนูที่ยอมให้ผู้ควบคุมสามารถเฝ้าดูกระบวนการและทำการควบคุมได้ทั้งการเปิดและปิด

2.Operator guide ให้คำแนะนำต่อผู้ควบคุม

-Plant-level Display เป็นระดับการแสดงผลที่อยู่ระดับบนสุดดังรูป 11 ลักษณะการแสดงผลจะเป็นการสรุปข้อมูลต่างๆ ที่จำเป็นสำหรับผู้ควบคุมเพื่อให้การควบคุมมีประสิทธิภาพมากขึ้นและแสดงถึงจุดที่ประสบปัญหาให้ผู้ควบคุมได้รับทราบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 13 ตัวอย่างการแสดงผลระดับกลุ่ม

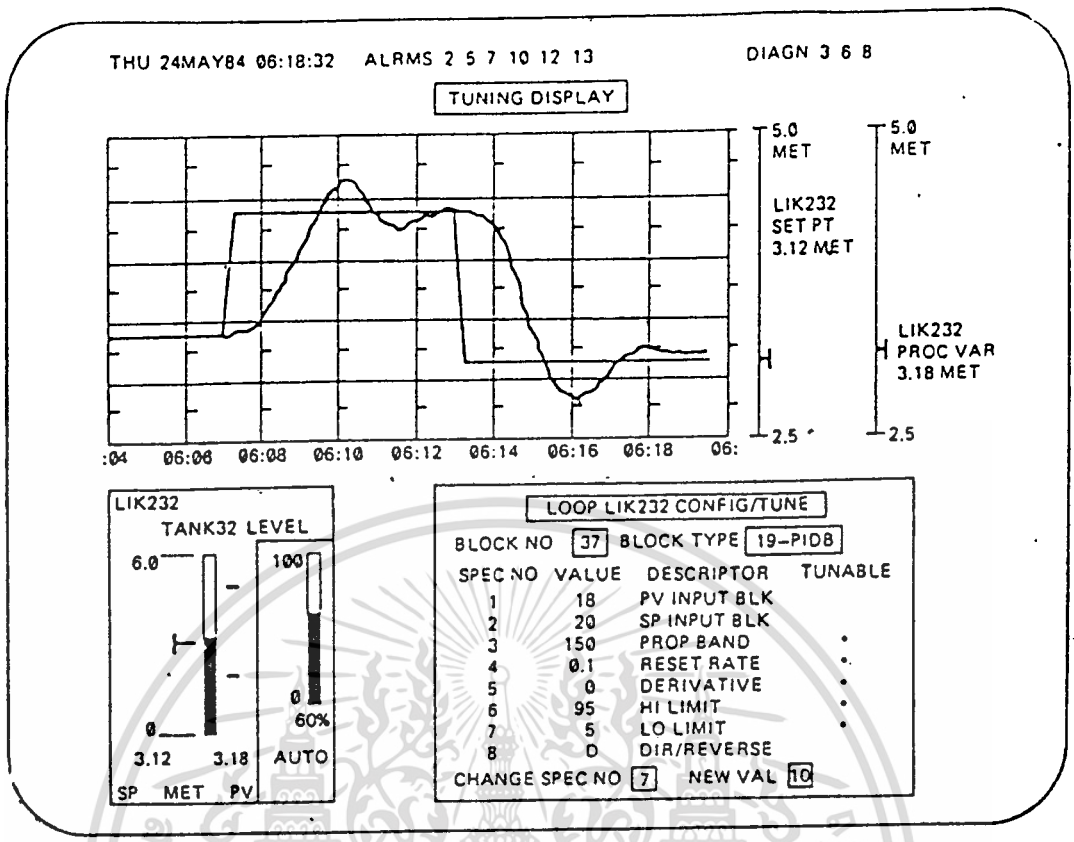
- Area-level Display เมื่อผู้ควบคุมได้ทราบลักษณะทั่วไปของกระบวนการผู้ควบคุมสามารถดูรายละเอียดข้างในได้อีก โดยลงมาดูใน plant area Loop-level Display ใช้งานได้ ถ้าหากผู้ควบคุมต้องการรายละเอียดเกี่ยวกับลูปควบคุมเพียงหน่วยเดียวก็ใช้การแสดงผลแบบลูป

รูป 14 แสดงตัวอย่างการแสดงผลแบบเทอร์นนิ่ง (turning) ซึ่งสามารถใช้ได้กับผู้ควบคุมและวิศวกรทางด้านวัดคุมฯ การแสดงผลแบบนี้มีองค์ประกอบช่วยในการเทอร์นนิ่ง ดังนี้

1. สามารถแสดงผลทางด้านแนวโน้มได้อย่างรวดเร็ว
2. สามารถใช้การควบคุมแบบด้วยมือหรือแบบอัตโนมัติ
3. แสดงค่าเทอร์นนิ่งของพารามิเตอร์ต่างๆ (P,I,D) ในลูป

ลักษณะกราฟแนวโน้มใช้แสดงจุดอ้างอิงเปลี่ยน (ในการควบคุมแบบอัตโนมัติ) และเอาท์พุทเปลี่ยนในการควบคุมด้วยมือ ผลตอบสนองที่ได้จะช่วยให้ผู้ควบคุมหรือวิศวกรวัดคุมฯ สามารถปรับตัวแปรต่างๆ ของตัวควบคุมได้ถูกต้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้ทำซ้ำ ดัดแปลง เผยแพร่ หรือการกระทำอื่นใดที่ก่อให้เกิดการละเมิดลิขสิทธิ์ ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 14 ตัวอย่างของการแสดงผลแบบเทอร์นึง

Graphic Display การแสดงผลในรูปแบบนี้อาศัยแผนภาพเข้าช่วย (P&ID) ผู้ควบคุมสามารถใช้การแสดงผลแบบนี้ในการควบคุมระดับพื้นที่ และระดับกลุ่ม ลักษณะโดยทั่วไปมีดังนี้

1.static field เป็นส่วนประกอบของตำแหน่งของไดนามิกเช่น รูปแท้งค์น้ำ รูปวาล์ว และอุปกรณ์อื่นที่ไม่มีการเปลี่ยนรูป

2.data field แสดงผลของข้อมูลที่มีการเปลี่ยนแปลงแบบอัตโนมัติ

3.dynamic display element เป็นการเปลี่ยนสี ขนาด รูปร่าง แสดงให้ทราบการเปลี่ยนแปลงของตัวแปรบางตัวในกระบวนการ

4.ความสามารถที่จะสร้างการแสดงผลกราฟฟิกซึ่งใช้เวลาหลายๆ ช่วงที่มากกว่าการแสดงผลแบบจอเดี่ยว ผู้ควบคุมสามารถเจาะลึกลงไปดูรายละเอียดที่ตัวเองสนใจ

<u>เงื่อนไขที่ใช้ในการออกแบบส่วนของแสดงผล</u>

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.การแสดงผลไม่จำเป็นต้องแสดงรายละเอียดในหน้าเดียวกัน เพียงแต่แสดงรายละเอียดที่สำคัญบางจุดก็พอ

2.การแสดงผลต้องดูแล้วน่ามองโดยสีไม่จ้าจนเกินไป

3. การพิจารณาข้างต้นนี้ บรรทัดแรกหรือ 2 บรรทัดแรกต้องช้อมูลธรรมดาที่น่าสนใจ ส่วน 2 บรรทัดท้ายใช้ในการสื่อสาร

4. สีที่ใช้ต้องทำให้ผู้ดูเกิดความสับสนน้อยที่สุด เช่น การใช้สีที่แน่นอนควรสงวนไว้กับส่วนที่อยู่กับที่, ส่วนที่เคลื่อนที่หรือข้อมูลเตือนภัย ถ้ากรณีเหล่านี้เป็นไปได้ก็ควรจะทำตาม ผู้ใช้ควรสามารถเลือกหรือเปลี่ยนสีในการแสดงผลทั้งแบบมาตรฐานและแบบที่เคยใช้กันเพื่อความจำเป็นในการประยุกต์

5. ไม่ควรใช้สีเป็นเพียงวิธีสำหรับการสื่อสารกับผู้ควบคุมในกรณีที่การทำงานนั้นสำคัญ เช่น การเตือนภัย เหตุการณ์ของการตาบอดสีของผู้ควบคุมมันมากเสียจนต้องยอมรับกรณีนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลักการทำงาน

หน้าที่หลักของหน่วยการติดต่อกับผู้ใช้ (Operator Inteface) คือ การนำเอาข้อมูลอันได้แก่ ค่าพารามิเตอร์ ค่าสถานะในกระบวนการผลิต ออกมาแสดงผลในรูปแบบต่าง ๆ ตามแต่ที่ผู้ใช้ต้องการ เนื่องจากโครงงานนี้เป็นการพัฒนาจากโครงงานเรื่องระบบควบคุมขบวนการด้วยคอมพิวเตอร์ ปีการศึกษา 2533 ดังนั้นจึงขอแสดงรายละเอียดของโครงสร้างข้อมูลเดิมที่เกี่ยวข้องกับโครงงานนี้ดังนี้

1) ข้อมูลที่เกี่ยวกับบล็อคการทำงาน มีการเก็บในรูปแบบ InstData อันประกอบด้วย

- InstType : ชนิดของบล็อคข้อมูล เช่น PIDC , LAG , ADD
- TagName :ชื่อของบล็อคซึ่งแยกย่อยเป็น ชื่อของบล็อค และหมายเลขประจำบล็อค
- DataField : ชนิดของข้อมูลภายในบล็อคที่ต้องการเข้าถึง เช่น MH , ML , PV
- Register และ VarAccess : ข้อมูลในบล็อคที่เกี่ยวกับการทำงานควบคุม

2) ข้อมูลเกี่ยวกับพอร์ต แบ่งเป็น 4 ชนิด คือ

- พอร์ตอินพุทอนาลอค
- พอร์ตเอาท์พุทอนาลอค
- พอร์ตอินพุทดิจิตอล
- พอร์ตเอาท์พุทดิจิตอล

ข้อมูลทั้งสองแบบมีรูปแบบการเข้าถึงข้อมูล (access) เป็นดังนี้

- ข้อมูลแบบบล็อค มีรูปแบบเป็น

IDDT(i_data [No.Block]) [Field]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

23

- ข้อมูลแบบพอร์ต มีรูปแบบดังนี้
  - พอร์ตอินพุทอนาลอค : ag_ipt [No.port]
  - พอร์ตเอาท์พุทอนาลอค : ag_opt [No.port]
  - พอร์ตอินพุทดิจิตอล : dg_ipt [No.port]
  - พอร์ตเอาท์พุทดิจิตอล : dg_opt [No.port]

ส่วนติดต่อกับผู้ใช้(OperatorInterface) ของโครงงานนี้ แบ่งเป็น 2 ลักษณะดังนี้

1) การแสดงพารามิเตอร์ในโหมดตัวอักษร

การแสดงผลแบบนี้มีการทำงานเป็น 3 ขั้นตอน คือ

1)การแสดงชื่อบล็อคทั้งหมดที่กำลังทำการควบคุมเพื่อทำการเลือกบล็อคที่ต้องการดูหรือแก้ไขค่า

2) การแสดงพารามิเตอร์ต่าง ๆ ในบล็อคที่เลือกเพื่อดู โดยสามารถทำการแก้ไขได้ถ้าบล็อคนั้นถูกเลือกโหมดการทำงานเป็นชนิด ENG

3) การแสดงค่าพอร์ต โดยจะแสดงค่าของพอร์ตที่เก็บอยู่ในบัฟเฟอร์ ณ ขณะที่เรียกดู

2) การแสดงผลในโหมดกราฟิก

เป็นการแสดงผลการทำงานของกระบวนการผลิตโดยใช้ภาพเคลื่อนไหว ซึ่งแบ่งการทำงานเป็น 2 ขั้นตอน

1) การสร้างภาพหลักของกระบวนการ (Static) : เป็นการสร้างภาพของกระบวนการโดยผู้ใช้โปรแกรมโดยผ่านซอฟท์แวร์ประเภท graphic editor ซึ่งในโครงงานนี้ใช้ PC PaintBrush

2) การกำหนดการแสดงผล : เมื่อผู้ใช้สร้างภาพหลักจากข้อ 1 เสร็จ ผู้ใช้จะต้องเรียกโปรแกรม GPBUILD.EXE เพื่อกำหนดตัวแปรที่ต้องการแสดงผล โดยการแบ่งตัวแปรการแสดงผลออกเป็น 4 แบบ ดังนี้

- แบบ ANALOG : เป็นการแสดงผลที่จะเคลื่อนไหว อยู่ภายในขอบ

เอกสารนี้เป็นลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ห้ามมิให้ทำซ้ำ ดัดแปลง หรือนำไปใช้เพื่อการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ลบข้อความนี้ออก และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เขตปิดที่กำหนดตามขนาดของสัญญาณที่อ่านมาจากกระบวนการผลิต สัญญาณที่สามารถนำมาแสดงในรูปแบบนี้ต้องเป็นสัญญาณ ประเภทอนาลอคที่มาจากบล็อค หรือพอร์ต

- แบบ DIGITAL : เป็นการแสดงผลตามค่าของสัญญาณชนิดดิจิตอลที่มาจากพอร์ต หรือจากบล็อค โดยเมื่อค่าสัญญาณเป็น 1 ตัวแปรภาพส่วนนี้ก็จะปรากฏ

- แบบ SIGNAL : เป็นการแสดงค่าของตัวแปรทุกรูปแบบ ที่เกี่ยวข้องกับกระบวนการผลิต โดยการกำหนดของผู้ใช้

- แบบ ALARM : เป็นการแสดงสภาวะของบล็อคที่ผู้ใช้กำหนด ว่ามีค่าผลการควบคุมเกินกว่าขอบเขตที่กำหนดหรือไม่

<u>ขีดความสามารถของโครงสร้างเก่า (CPCS)</u>

-ทำงานด้วย Time slicing Multitasking Real-time Operating System

-มีความสามารถในด้านแสดงผลด้านต่างๆ เช่น Instrument Calibration, Process Trending and alarm display เป็นต้น

-ขีดความสามารถเหมาะกับขบวนการที่เป็น Slow Process

-การปรับค่าพารามิเตอร์ต่างๆ ของการควบคุมตลอดจนการควบคุมการทำงานจะกระทำผ่านคีย์บอร์ดและสังเกตผลตอบสนองของขบวนการต่างๆ ผ่านทาง operator console

แนวความคิดเกี่ยวกับโครงสร้างของ CPCS

1. อุปกรณ์การคำนวณ (Control Instrument) ทุกชนิดอยู่ในรูปบล็อก บล็อกต่างๆ มีความสามารถในการคำนวณในรูปแบบที่ต่างๆ กัน และถูกนำมาต่อถึงกันโดยอาศัยสัญญาณอ้างอิงถึงกัน (Signal Addressing)

Signal Addressing จะบอกถึงชนิดและแหล่งที่มาของสัญญาณโดยมีรูปแบบดังนี้

tt : ssssssss โดย tt เป็น signal type

และ ssssssss เป็น signal source

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารนี้ทุกครั้งที่มีการนำไปใช้

2. มีความยืดหยุ่นสำหรับการควบคุมแบบต่างๆ ฟังก์ชั่นการทำงานของแต่ละขบวนการ จะได้จากการติดตั้งของผู้ใช้

ส่วนประกอบของ CPCS

Control Utilities ของ CPCS ของโครงงานเก่าประกอบด้วย

-Control Calibration and Instrument Face Plate (Control Group)

-Process Trending

-Process alarm

ผู้ใช้สามารถติดตั้ง Utilities ในระดับที่ต้องการได้ ต่อไปนี้จะกล่าวถึงระบบทั้งหมดของ CPCS

ระบบ CPCS ประกอบด้วย 2 ส่วนด้วยกัน คือ

1.Controller : เป็นส่วนที่ควบคุมขบวนการตาม Configuration ที่ผู้ใช้ได้ตั้งขึ้น รวมทั้งแสดงยูทิลิตี้ต่างๆ สำหรับผู้ควบคุมได้ใช้ประโยชน์

2.Builder : เป็นส่วนที่สร้าง Configuration สำหรับ Controller โดย Configuration หลายอันเหล่านี้ประกอบด้วย

-Control instrument ต่างๆ

-การออกแบบการแสดงผลของแต่ละยูทิลิตี้

ข้อกำหนดการใช้โปรแกรม

โปรแกรมการสร้างข้อมูลควบคุม (BUILDER.EXE)

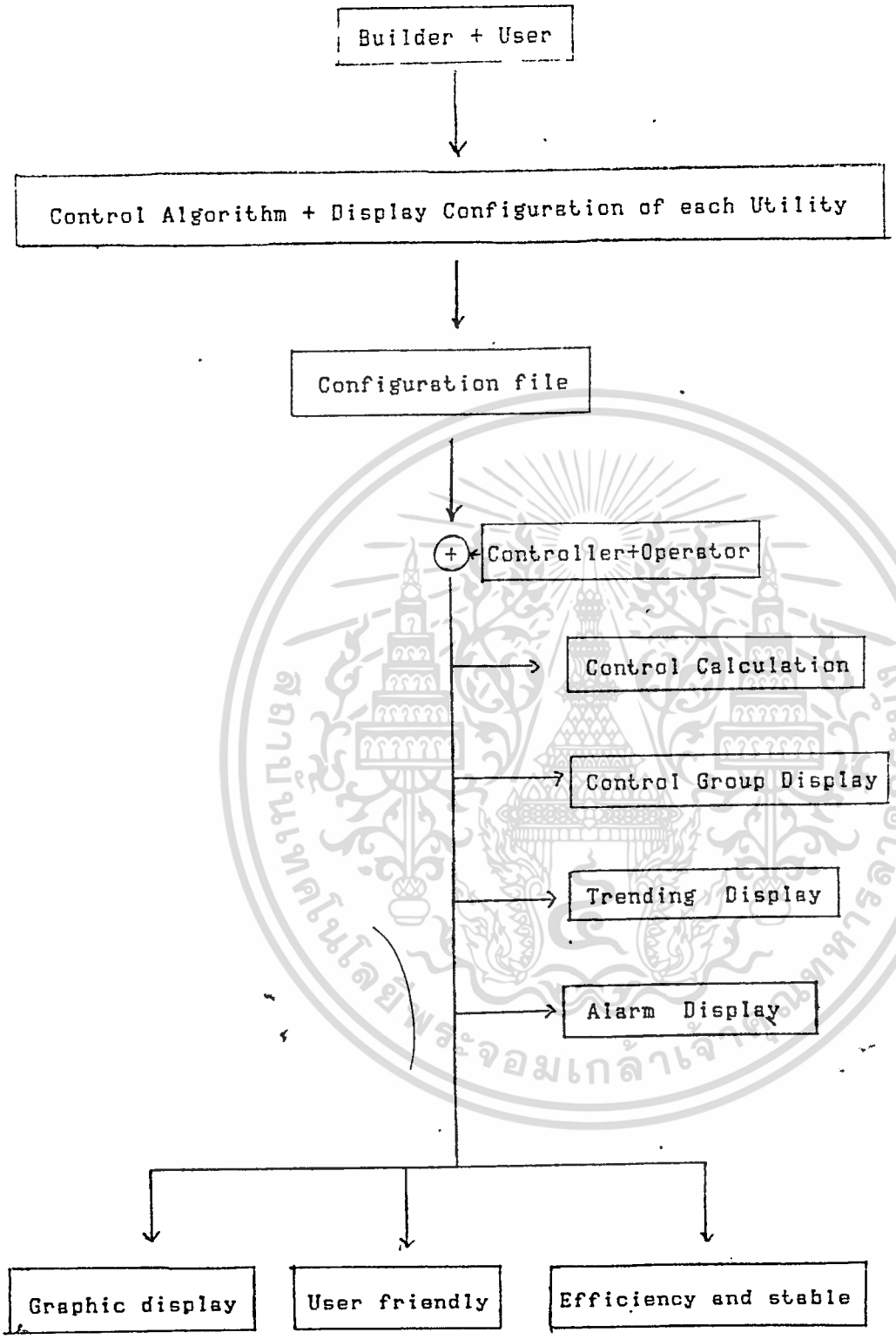
- เครื่องคอมพิวเตอร์ทั่วไปที่มีหน่วยความจำเหลือก่อนรันโปรแกรม 320 kB
- เม้าส์พร้อมไดร์ฟเวอร์
- จอภาพสีระดับ VGA ที่มีความละเอียดอย่างต่ำขนาด 640x350 จุด
- ในไดเรคทอรี่เดียวกับโปรแกรมต้องมีไฟล์ดังนี้

1)CPCCFG.DAT 2)CPCPIC.PCX

โปรแกรมการควบคุม (CTRLLER.EXE)

- เครื่องคอมพิวเตอร์ระดับ 80386 ที่มีหน่วยความจำเหลือก่อนรันโปรแกรม 450 kB
- จอภาพสีระดับ VGA ที่มีความละเอียดอย่างต่ำขนาด 640x350 จุด
- ในไดเรคทอรี่เดียวกับโปรแกรมต้องมีไฟล์ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปแสดงส่วนประกอบต่างๆ ของระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1)CPC.FNT 2)CPCE.FNT 3)STND.FNT 4)STNDE.FNT

5)CPCCFG.DAT 6)GPCFG.DAT 7)CPCPIC.PCX

การใช้งานซอฟท์แวร์

เนื่องจากเป็นโครงงานต่อเนื่อง โครงงานนี้จึงพยายามออกแบบการใช้งานให้ใกล้เคียงกับโครงงานเดิมให้มาก โดยเฉพาะคีย์การใช้งาน และลักษณะการใส่ชื่อพารามิเตอร์ต่างๆ ที่ต้องการเข้าถึง สำหรับส่วนการทำงานที่เพิ่มขึ้นนั้นมีวิธีการใช้งานดังนี้

การแสดงผลในโหมดตัวอักษร

การทำงานในโหมดนี้เริ่มทำงานเมื่อกดคีย์ F10 ขณะที่โปรแกรมกำลังทำงานควบคุมกระบวนการอยู่ ไม่ว่ากำลังอยู่ในโหมดการแสดงผลใดก็ตาม คีย์ F10 นี้เป็นฮอตเก้ีลคีย์สำหรับการแสดงผลโหมดนี้ คือ สามารถเลิกการแสดงผลได้ทันทีโดยการกด F10 อีกครั้ง

หน้าจอแรกที่เห็นเมื่อเข้าทำงานในโหมดนี้ จะเป็นการแสดงชื่อบล็อคที่โปรแกรมกำลังทำงานอยู่ โดยจะมี SIGANL เป็นตัวสุดท้าย เราสามารถเลื่อนคีย์ลูกศรซ้ายขวาไปยังชื่อบล็อคที่เราต้องการ ถ้าเลื่อนไปที่ SIGNAL จะหมายถึงการดูค่าเกี่ยวกับพอร์ต เมื่อได้ตำแหน่งที่ต้องการ จึงกดคีย์ Enter เพื่อเข้าสู่หน้าจอถัดไป

หน้าแสดงผลเกี่ยวกับบล็อค จะแสดงค่าพารามิเตอร์ทุกตัวที่เกี่ยวกับบล็อคนั้น ซึ่งสามารถทำการแก้ไขได้ ถ้าบล็อคนั้นถูกออกแบบมาให้แก้ไขได้ การแก้ไขสามารถทำได้โดยการเลื่อนแถบสีสว่างไปยังค่าพารามิเตอร์นั้น แล้วกดคีย์ Enter จากนั้นใส่ค่าที่ต้องการลงไปแล้วกดคีย์ Enter อีกครั้งโปรแกรมจะนำค่าใหม่ไปทำงาน

สำหรับหน้าจอแสดงผลเกี่ยวกับพอร์ตจะมีลักษณะการใช้งานคล้ายกัน แต่การเปลี่ยนแปลงค่าสามารถทำได้เฉพาะพอร์ตเอาท์พุทเท่านั้น

เมื่อเราใส่ค่าพารามิเตอร์ใหม่เรียบร้อย การแสดงผลโหมดนี้จะหยุดทำงานและกลับไปที่โหมดการแสดงผลเดิม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.Controller : เป็นส่วนที่ควบคุมขบวนการตาม Configuration ที่ผู้ใช้ได้ตั้งขึ้น รวมทั้งแสดงยูทิลิตี้ต่างๆ สำหรับผู้ควบคุมได้ใช้ประโยชน์

2.Builder : เป็นส่วนที่สร้าง Configuration สำหรับ Controller โดย Configuration หลายอันเหล่านี้ประกอบด้วย

-Control instrument ต่างๆ

-การออกแบบการแสดงผลของแต่ละยูทิลิตี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หมายเหตุ ค่าพารามิเตอร์ต่างๆ ที่ถูกนำมาแสดง เป็นค่าที่อยู่ในบัฟเฟอร์ของโปรแกรม ขณะที่ทำการเรียกดู

การแสดงผลในโหมดกราฟิก

แม้ว่าการทำงานของทาก์สนี้ในโปรแกรมการควบคุมยังไม่สามารถใช้งานได้ แต่ก็สามารถอธิบายขั้นตอนการทำงานได้ดังนี้

การแสดงผลในโหมดกราฟิก แบ่งเป็น 3 ขั้นตอน ดังนี้

1) การออกแบบภาพ สามารถทำได้โดยใช้โปรแกรม PC PaintBrush เป็น Graphic Editor แล้วเก็บภาพลงในไฟล์ชื่อ CPCPIC.PCX โดยไฟล์ภาพต้องอยู่ในไดเรคตอรี่เดียวกันกับโปรแกรม CTRLLER

2) การกำหนดรูปแบบการแสดงผล การทำงานในส่วนนี้จะเริ่มด้วยการเรียกส่วน GRAPHIC ในโปรแกรม BUILDER.EXE ซึ่งโปรแกรมนี้จะเรียกภาพจากไฟล์ CPCPIC.PCX มาแสดงเพื่อให้ผู้ใช้กำหนดสิ่งที่ต้องการแสดงผล รูปแบบการแสดงผลแบ่งเป็น 4 ลักษณะ คือ ANALOG , DIGITAL , SIGNAL และ ALARM

3) การแสดงผลในโหมดนี้จะเป็นอีกทาก์สหนึ่งเช่นเดียวกับ FacePlate Trending และ Alarm ซึ่งสามารถเลือกการแสดงผลโดยการใช้คีย์ F1 หรือ F2

ขั้นตอนการสร้างการแสดงผล

1) ใช้คีย์ F4 และ F5 เพื่อเลือกรูปแบบการแสดงผลและสี

2) ใส่ชื่อของสัญญาณที่ต้องการแสดงผล โดยมีรายละเอียดดังนี้

กรณีแสดงผลแบบ ANALOG : ชื่อสัญญาณจะเป็นพอร์ตชนิดอนาลอคหรือชื่อสัญญาณอนาลอคในบล็อค เช่น PV PID1 , MV PID2

กรณีแสดงผลแบบ DIGITAL: ชื่อสัญญาณจะเป็นพอร์ตชนิดดิจิตอลหรือชื่อสัญญาณดิจิตอลในบล็อค

กรณีแสดงผลแบบ SIGNAL : ชื่อสัญญาณจะเป็นพอร์ตชนิดใดหรือชื่อสัญญาณในบล็อคชนิดใดก็ได้

กรณีแสดงผลแบบ ALARM : ชื่อที่ใส่จะเป็นชื่อของบล็อคที่มีการกำหนดขอบเขตการทำงาน เช่น บล็อคชนิด PIDC , NONL

เอกสารนี้เป็นเอกสารสงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น ไม่อนุญาตให้ทำซ้ำ ดัดแปลงเนื้อหา หรือนำไปเผยแพร่ต่อ เพื่อการค้า ในทุกรูปแบบที่มีการนำไปใช้

กรณีที่ผู้ใช้ใส่ชื่อสัญญาณที่ไม่สอดคล้องกับลักษณะการแสดงผลหรือใส่ชื่อสัญญาณผิด โปรแกรมจะให้ใส่ชื่อสัญญาณใหม่

3) เมื่อกำหนดชื่อสัญญาณเรียบร้อย กรณีการแสดงผลแบบ SIGNAL และ ALARM ให้ผู้ใช้ใช้เมาส์เลื่อนไปกำหนดตำแหน่งที่ต้องการแสดงผล แล้วกดปุ่มซ้ายของเมาส์ ในกรณีการแสดงผลแบบ DIGITAL การแสดงผลจะแสดงเป็นกรอบสี่ ผู้ใช้จึงต้องกำหนดขนาดและตำแหน่งของกรอบที่ต้องการแสดงโดยกำหนดตำแหน่งมุมบนซ้ายและมุมล่างขวาของกรอบโดยกดปุ่มซ้ายของเมาส์ สำหรับในกรณีการแสดงผลแบบ ANALOG นั้น ผู้ใช้ต้องกำหนดจุดสองจุดภายในรอบวงปิดที่ต้องการแสดงผล เช่น วงกลม วงรี หรือ สี่เหลี่ยม โดยจุดแรกที่กำหนดจะเป็นขอบเขตการแสดงผลที่สูงที่สุดในรูปวงปิดนั้นเมื่อสัญญาณมีค่าเป็น 100% ส่วนจุดที่สองเป็นขอบขต่ล่างสุดที่ต้องการแสดงผลเมื่อสัญญาณมีค่าเป็น 0% โดยจุดแรกต้องอยู่ทางซ้ายมือของจุดขวาเสมอ

4) เมื่อกำหนดค่าต่าง ๆ เสร็จ โปรแกรมจะแสดงภาพการแสดงผลออกมา ถ้าการแสดงผลไม่เป็นที่พอใจ ผลใช้สามารถยกเลิกการแสดงผลที่สร้างขึ้นโดยการกดคีย์ ESC เพื่อย้อนกลับไปป้อนข้อมูลใหม่

5) เมื่อต้องการเลิกการทำงานหรือเสร็จสิ้นการกำหนดข้อมูล ทำได้โดยการกดคีย์ F10

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รายละเอียดการทำงาน

การทำงานของซอฟแวร์ตัวนี้มีด้วยกันหลายส่วน แต่จะขอยกเอาส่วนที่สำคัญต่อการทำงานมาอธิบายซึ่งมีอยู่ 3 ส่วน ดังนี้

- การเพิ่มส่วนการทำงาน(task)
- การเข้าถึงและแก้ไขข้อมูลในการควบคุม
- การสร้างและเรียกภาพกราฟิกออกมาใช้งาน

การเพิ่มส่วนการทำงาน

การทำงานของมัลติทาสค์กิ้ง(Multitasking)

เมื่อเอ่ยถึงคำว่า "Multitasking" ขึ้นมาโดยทั่วไปจะนึกถึงการทำงานของหลาย ๆ โปรแกรมบนระบบเดียวกันโดยอาศัยระบบปฏิบัติการ (OS) เช่น OS/2 ,UNIX เข้าช่วยในการจัดการและกำหนดงานที่ต้องทำแต่อย่างไรก็ตามยังมีวิธีหนึ่งที่ใช้ในการเขียนโปรแกรมมัลติทาสค์กิ้ง โดยปราศจากการช่วยเหลือของระบบปฏิบัติการ เราเรียกว่า Muititasking kernel

Muititasking kernel เราเขียนโดยเทอร์โบ ซี (Turbo C) และทำงานภายใต้ดอส(DOS) สิ่งที่จะขาดไม่ได้สำหรับระบบมัลติทาสค์กิ้งคือ รูทีน(routine) ที่ทำหน้าที่ในการเลือกงานต่าง ๆ เราเรียกว่า scheduler ซึ่งเป็นรูทีนที่ทำการเปลี่ยนค่าแสตกพอยน์เตอร์(stack pointer) จากค่าเก่าไปเป็นค่าใหม่ ที่เป็นที่เก็บค่าอินสตรัคชั่นพอยน์เตอร์(IP) ของอีกงานหนึ่งและในแต่ละงานมีแสตกของมันเอง

ตัวอย่างของมัลติทาสค์กิ้งแบบง่าย ๆ เพื่อประกอบความเข้าใจ

ในระบบมีงาน A และ B ให้ A กำลังงานอยู่ เมื่อทาสค์สวิทช์(scheduler) แอกทีฟขึ้นโดยเกิดจากการสลับเข้ามา สถานะของงาน A ถูกเก็บลงแสตกและค่าของแสตกพอยน์เตอร์จะถูกกำหนดให้ชี้ไปที่แสตกของ B และเมื่อจบการทำงานของทาสค์สวิทช์แล้วก็จะเริ่มการทำงานของงาน B ต่อไป เมื่อทาสค์สวิทช์แอกทีฟอีก สถานะของ B จะถูกเก็บลงแสตกและแสตกพอยน์เตอร์จะถูกกำหนดให้ไปชี้ที่ส่วนบนของแสตกของ A เมื่อทาสค์สวิทช์จบการทำงานก็จะเริ่มการทำงานของ A ต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในโครงงาน Digital Control System จะใช้ระบบมัลติทาสค์กิ้งเป็นหัวใจหลักของการทำงาน โดยส่วนประกอบของมัลติทาสค์กิ้งอยู่ในไฟล์ cmtsk.c รายละ

เอียดของการทำงานในแต่ละฟังก์ชั่นเป็นดังนี้

1.int8_task_switch() เป็นฟังก์ชั่นที่เลือกฟังก์ชั่นที่จะทำงานต่อไปได้แก่ task_switch หรือinterval_task ภายในฟังก์ชั่นนี้จะมีส่วนประกอบคือ ส่วนที่เก็บแสตกเซคเมนท์และแสตกพอยน์เตอร์ลงในบัฟเฟอร์ ตัวนับ ว่าจะถึงเวลาที่จะเรียก interval_task มาทำงาน, คืนค่าของแสตกเซคเมนท์และแสตกพอยน์เตอร์ให้กับงานที่พร้อมที่จะทำงาน

2.task_switch() เป็นฟังก์ชั่นที่ทำการเลือกงาน Manager, Keyboard, Timebase และงานอื่น ๆ ที่พร้อมที่จะทำงานได้แก่ Trending,Face Plate, Alarm, Graphic ซึ่งงานทั้ง 4 อย่างนี้ จะทำงานก็ต่อเมื่อกดคีย์ให้ทำงานเท่านั้น

ส่วนประกอบของฟังก์ชั่นนี้

1.ตรวจสอบสถานภาพของงานต่างๆ ว่าตายอยู่หรือเปล่า โดยจะเรียก all_dead()

2.หางานที่พร้อมจะทำงานได้ หมายถึงสถานะของงานอยู่ในสภาพ Ready ถ้าพบก็จะไปเรียก auto_switch()

3.interval_task() เป็นงานที่จะถูกเรียกเมื่อตัวนับใน int8_task_switch() ถึงค่าที่ตั้งไว้ ฟังก์ชั่นนี้จะไปเรียก user_interrupt() และ manual_switch() สรุปแล้วจะเลือก
งานที่เกี่ยวกับการคำนวณมาทำงาน

4.auto_switch() ถูกเรียกโดย task_switch() หน้าที่ของฟังก์ชั่นนี้คือ คืนค่าแสตกพอยน์เตอร์และแสตกเซคเมนท์ ให้แก่งานที่พร้อมจะทำงาน

5.manual_switch() ถูกเรียกโดย interval_task() ลักษณะการทำงานนี้จะคล้าย auto_switch()

6.all_dead() คอยตรวจสถานะว่าอยู่ในสภาพที่ตายอยู่หรือเปล่า

7.all_suspended() ตรวจสอบสถานะของทุกงานว่าอยู่ในสภาพหยุดการทำงานหรือเปล่า

8.all_sleeping() ตรวจสอบสถานะของทุกงานว่าอยู่ในสภาพหลับอยู่หรือเปล่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

9.free_all() คือค่าแสตกคืน

10.make_task() จองหน่วยความจำสำหรับงานต่างๆ เพื่อเก็บสถานะของรีจิส

เตอร์ต่างๆ ที่มีการเปลี่ยนงานเกิดขึ้น

11.multitask() เริ่มต้นการทำงานของมัลติทาสค์กิ้ง นอกจากยังจองหน่วยความจำสำหรับ task_switch() กับ interval_task()

12.kill_task() ทำให้สถานะของงานอยู่ในสภาพที่ตาย

13.stop_tasking() ออกจากการทำงานของมัลติทาสค์กิ้งโดยจะเรียก manual_switch()

14.mono_task() เปลี่ยนจากการทำงานของมัลติทาสค์กิ้งให้อยู่ในรูปของงานเดี่ยวๆ

15.setsem() ทำหน้าที่จองทรัพยากรส่วนกลางเพื่อกันการใช้งานซ้ำซ้อน

16.clrsem() คืนทรัพยากรส่วนกลางกลับคืนเพื่อให้งานอื่นๆ สามารถใช้งานได้

17.init_task() กำหนดลักษณะเริ่มต้นของงานต่างๆ

18.send() ทำหน้าที่ในการส่งข่าวสารระหว่างฟังก์ชั่นแต่ละฟังก์ชั่น

19.recv() ทำหน้าที่ในการรับข่าวสารระหว่างฟังก์ชั่นแต่ละฟังก์ชั่น

20.resume() ทำให้งานต่างๆ พร้อมที่จะเริ่มทำงานได้

21.suspend() ทำให้งานอยู่ในสถานะหยุดค้างชั่วคราว

<u>การเพิ่มทาค์ก</u>

1. ฟังก์ชั่น RunGraphic() และ StopGraphic() ลงในไฟล์ ocmd.c โดยฟังก์ชั่นนี้มีโครงสร้างเหมือนกับฟังก์ชั่น RunTrending() และ StopTrending() โดยมีลักษณะดังนี้

```
void RunGraphic(void)
        {  message   sm;
             sm.did = GprunID;
             sm.sid = ManagerID ;
             sm.s   = 'R' ;
             sm.fn  = WAIT;
             resume(GprunID);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        while(send(&sm)==ERR);
        gprun =1;
    }

void StopGraphic(void)
    {  message   sm;
        gprun =0;
        sm.did = GprunID;
        sm.sid = ManagerID ;
        sm.s   = 'S' ;
        sm.fn  = WAIT;
        resume(GprunID);
        while(send(&sm)==ERR);
    }
```

2. เพิ่มฟังก์ชั่นStopGraphic() และ การตรวจสอบ gprun เข้าไปในส่วนของการตรวจสอบคีย์ของทุกคีย์

3. เพิ่มการตรวจสอบคีย์ของ Graphic ใน manager()

```
    if(rm.s =='G'
        {
          if(fprun) StopTrend();
          if(fprun) StopFacePlate();
          if(alarm) StopAlarm();
          RunGraphic();
          continue;
        }
```

4. เพิ่มฟังก์ชั่นStopGraphic() และ การตรวจสอบ gprun เข้าไปในส่วนของ StopCPC()

5. เพิ่มจำนวนทาค์ก (MAX_TASK) ให้มากขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การเข้าถึงและการแก้ไขข้อมูล

ข้อมูลที่ใช้อยู่ในโปรแกรมควบคุม มีอยู่ด้วยกันหลายแบบ ซึ่งมีความสอดคล้องกันอยู่ ในที่นี้จะแสดงการเข้าถึงข้อมูลเฉพาะข้อมูลที่เกี่ยวข้องกับการแสดงค่า

การแสดงชื่อบล็อค

```
IDTG(i_data[indx_tag])
```

การแสดงค่าพอร์ต

```
                dg_ipt[i] ? 1:0     สำหรับค่าดิจิตอลอินพุทพอร์ต
              numfmt)ag_ipt[i])     สำหรับค่าอนาลอกอินพุทพอร์ต
          DABF(dg_opt[indx_tag])    สำหรับดิจิตอลเอาท์พุทพอร์ต
numfmt(DABF(ag_opt[indx_tag]))      สำหรับอนาลอกเอาท์พุทพอร์ต
```

การแสดงชนิดของบล็อค

```
inst[IDTP[indx_tag]]
```

การแสดงสถานะของบล็อค

```
ISSTATUS((IDDT(i-data[indx_tag])[IS],t)
```

การแสดงโหมดของบล็อค

```
imode[IDDT(i_data[indx_tag])[MD]]
```

การแสดงลูปของบล็อค

```
iloop[IDDT(i_data[indx_tag])[LP]]
```

การแสดงพารามิเตอร์อื่นๆของบล็อค

```
numfmt(IDDT(i_data[indx_tag])[xx])
```

xx : เป็นชนิดของพารามิเตอร์ที่ต้องการโดยอ้างอิงจาก enum field;

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การกำหนดการเข้าถึงพารามิเตอร์ในซอฟท์แวร์

จากหัวข้อที่แล้วการเข้าถึงข้อมูลต่างๆ ในการควบคุมไม่ว่าจะเป็นพอร์ตหรือบล็อค เราจะสังเกตได้ว่าตัวแปรที่ใช้ในการเข้าถึงนั้นมี 2 แบบ คือ

- ตัวแปรที่บอกว่าเป็นพอร์ตชนิดใด หรือเป็นพารามิเตอร์ใดของบล็อค และ
- ตัวแปรที่บอกหมายเลขของพอร์ต หรือหมายเลขของบล็อค ซึ่งในการทำงานควบคุมทั้งสองตัวแปรนี้จะถูกเก็บในรูปของ VarAccess ซึ่งมีโครงสร้างดังนี้

```
struct
{  Field  fld_d;
   int    ary_d;
} VarAccess;
```

ดังนั้นเมื่อผู้ใช้กำหนดชื่อของบล็อคหรือพอร์ตและพารามิเตอร์ลงไป ซอฟแวร์จะทำการแปลงให้สิ่งที่ผู้ใช้คีย์ไปเป็นในรูป VarAccess โดยใช้ฟังก์ชัน

```
Search_Inst( );
Search_Port( );
str_dfld( );
```

การสร้างภาพ Static และการนำไปใช้

การสร้างส่วน Static นั้นเป็นรายละเอียดที่ผู้ใช้สามารถกำหนดได้โดยอิสระในด้านรูปร่าง ยกเว้นเรื่องของสีในบางตำแหน่งที่จะกล่าวภายหลัง ในโครงงานนี้ได้เลือกใช้ PC PaintBrush version 3.35 เป็น graphic editor เนื่องจาก

1) โปรแกรมเป็นที่รู้จักกันกว้างขวาง การใช้งานทำได้ง่าย สามารถเรียนรู้ได้เร็ว
2) โปรแกรมมีขนาดเล็กเมื่อเทียบกับโปรแกรมตัวอื่น
3) ในตัวโปรแกรมเองมีไลบรารีของสัญลักษณ์ที่เกี่ยวกับการควบคุมกระบวนการเรียบร้อยอยู่แล้ว

เอกสารนี้เป็นเอกสาร... ไม่ว่ากรณีใด... ห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การเก็บภาพที่สร้างเสร็จแล้วของ PC PaintBrush นี้ จะไม่เหมือนโปร

แกรมอื่น เช่นAutoCAD ซึ่งเก็บภาพเป็นรหัส เพราะ PC PaintBrush เก็บรูปลงไฟล์ในลักษณะไฟล์สกุล.pcx ซึ่งเป็นการเข้ารหัสข้อมูลภาพ โดยใช้วิธี RLE(Run-Length Encoding)

<u>การเก็บภาพลงไฟล์สกุล .pcx</u>

ภาพที่เราสร้างขึ้นจะประกอบด้วยจุด (pixel) มากมาย ซึ่งจะมีสีต่างกันไปตามการสร้างของผู้ใช้ ข้อมูลของจุดในตำแหน่งต่างๆ ของภาพจะถูกอ่านจากแนวนอนจากซ้ายไปขวา และจากบนลงล่างจนหมดภาพ แล้วนำข้อมูลเหล่านี้ไปเข้ารหัสเพื่อลดขนาดไฟล์ ซึ่งมีรายละเอียดดังตารางนี้

เงื่อนไข	การกระทำ
ค่าที่เหมือนไบต์อื่น	นับ Counter + 1
ค่า < 'C0h' ไม่เหมือนไบต์อื่น	เก็บลงไฟล์เลย
ค่า > 'C0h', ไม่เหมือนไบต์อื่น	เก็บ 'C1h' ลงไฟล์ตามด้วยค่าไบต์นั้น
Counter > 63	Counter=1, เก็บค่า 'FFh' และค่าไบต์ลงไฟล์

ตารางที่ 2 แสดงเงื่อนไขการเข้ารหัสของไฟล์สกุล.pcx

ถ้าไบต์ใดๆมีค่าไม่เหมือนไบต์อื่นๆ โดยที่ค่า 2 บิตบนไม่เท่ากับ "11"ไบต์นั้นจะถูกเก็บลงไฟล์ โดยจะมีตัวนับว่ามีเหมือนกันกี่ไบต์ แต่ต้องไม่เกิน 63 ถ้าเกินให้นำมาที่ตัวนับที่ได้ออร์ (OR) กับค่า "C0H" แล้วเก็บมาตัวนับลงไฟล์ แล้วตามด้วยค่าไบต์นั้นๆ จากนั้นเริ่มต้นนับเป็น 1 ใหม่ แต่ถ้าค่าบิตบนเป็น "11" ให้เขียนตัวนับ = 1 ลงในไฟล์แล้วตามด้วยค่าไบต์นั้น

โครงสร้างไฟล์สกุล .pcx จะสงวน 128 ไบต์แรกของไฟล์ สำหรับเก็บข้อมูลต่างๆในการถอดรหัสไฟล์ โดยมีรายละเอียดตามตารางที่ 3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไบต์ที่	ขนาด (Byte)	ชื่อข้อมูล	รายละเอียด
0	1	Password	ค่า '0aH' = ไฟล์ .PCX
1	1	Version	เก็บค่า Version (รุ่น) ของ PC Paintbrush ดังนี้ 0=Version 2.5 2=Version 2.8 with Palette 3=Version 2.8 without Palette 5=Version 3.0
2	1	Encoding	ค่า = 1
3	1	Bits per Pixel	จำนวนของบิตที่ใช้แสดง 1 Pixel จาก 1 Plane (ระนาบสี) 1 = EGA, VGA, or HERC 4 = CGA
4	8	Windows-Dimemsions	ค่า integer 4 ค่า (ค่าละ 2 Byte) ให้ค่ามุมบนซ้ายและมุมล่างขวาของภาพ กำหนดให้รูปแบบ x1, y1, x2, y2
12	2	Horizontal-Resolution	ความละเอียดของการแสดงภาพในแนวนอน 640 = EGA, VGA 320 = CGA 720 = HERCULES
14	2	Vertical-Resolution	ความละเอียดของการแสดงภาพในแนวตั้ง 480 = VGA 350 = EGA 200 = CGA 348 = HERC
16	48	Color Map	ข้อมูล Color Palette ดูรูป 2
64	1	Reserved	
65	1	No. of Plane	จำนวนของ Planes ที่ใช้แสดงภาพ
66	1	Byte per Line	จำนวนไบต์ต่อการสแกน 1 บรรทัด
68	2	Palette Info	How to interpret the palette
70	8	Maximum x value	Special for fractal files
78	8	Minimum x value	Special for fractal files
86	8	Maximum y value	Special for fractal files
94	8	Minimum y value	Special for fractal files
102	8	P Value	Special for fractal files
110	8	Q Value	Special for fractal files
118	10	Not used	

หมายเหตุ ข้อมูลไบต์ที่ 70-127 ใช้เฉพาะ Fractal Programming

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3 แสดงโครงสร้างไฟล์สกุล .pcx ใน 128ไบต์แรก

การถอดรหัสไฟล์สกุล .pcx

การถอดรหัสก็กระทำย้อนกลับกับเข้ารหัส โดยเริ่มจากการอ่านข้อมูล 1 ไบต์มาดูว่ามากกว่า COH หรือไม่ (ตรวจสอบว่าสองบิตบนเป็น 11) ถ้าใช่ค่านั้นเป็นค่าตัวนับโดยการนำค่านั้นมา XOR กับ COH จะได้ค่าตัวนับออกมาไบต์ต่อไปก็จะเป็นค่าของไบต์ที่นำมาขยาย แต่ถ้าค่าที่อ่านน้อยกว่า COH จะทำการเก็บลงหน่วยความจำเลย โดยไม่นำมาขยาย จะทำเช่นนี้จนจบไฟล์ จากนั้นนำข้อมูลในหน่วยความจำมาแสดงภาพโดยใช้คำสั่ง POKEB( );

อัลกอริธึมของการถอดรหัส

```
loop  until EOF
read byte n
 if byte n > COH
  read next byte n+1
  decode byte n+1 for byte n
end
```

ลักษณะการเก็บภาพของไฟล์สกุล .pcx สามารถทำได้ทั้งในรูปจอสีเดียวและจอสีระดับต่างๆ แต่เพื่อให้การทำงานของการเก็บและแสดงภาพสอดคล้องกับโปรแกรมจึงกำหนดการแสดงผลของโปรแกรม PC PAintBrush เป็นจอระดับVGA 16สี ความละเอียด 640 * 350 จุด ดังนั้นฟังก์ชันDrawStatic ( ); จึงกำหนดการทำงานเข้ากับข้อกำหนดนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การแสดงภาพ

เพื่อให้การแสดงภาพ Static รวดเร็ว จึงได้มีการเก็บภาพที่อ่านจากไฟล์

มาเก็บในหน่วยความจำของเครื่อง แทนการอ่านภาพจากไฟล์ โดยใช้ฟังก์ชัน KeepStatic (); ซึ่งจะจองเนื้อที่หน่วยความจำเป็นสองส่วนเพื่อดักเก็บภาพที่ได้จากฟังก์ชัน DrawStatic(); ลงในหน่วยความจำโดยใช้คำสั่ง getimage(); จากนั้นเมื่อต้องการแสดงภาพจะใช้คำสั่ง putimage (); เรียกภาพมาขึ้นจอ เหตุผลที่ต้องแบ่งภาพเป็นสองส่วน (up , down) เนื่องจากข้อกำหนดจากการทำงานของ Turbo C ที่ยอมรับการจองเนื้อที่หน่วยความจำได้ไม่เกิน 64 KB แต่ภาพบนจอขนาด 640*350 จุด 16 สี มีขนาดประมาณ 112 KB จึงมีการแบ่งภาพเป็นสองส่วนเพื่อเก็บลงหน่วยความจำ

ข้อจำกัดเกี่ยวกับภาพ Static

เนื่องจากซอฟท์แวร์ตัวนี้เขียนโดยใช้ Turbo C ดังนั้นข้อจำกัดต่างๆ เกี่ยวกับการสร้างภาพหลักนั้น มีผลสืบเนื่องมาจากความจำกัดด้านกราฟิกของ Turbo C

กราฟิกของ Turbo C จำกัดการแสดงสีหลักไว้ 16 สี ดังนั้นในการสร้างภาพหลักนั้นผู้ใช้ต้องพิจารณาในส่วนนี้เสมอ ต้ว PC PaintBrush เอง มีความสามารถในการแสดงสีได้มากกว่า Turbo C ดังนั้นการกำหนดสีของบางสิ่งในรูปต้องอยู่ในขอบเขตที่ Turbo C สามารถทำได้ เช่น กำหนดการแสดง alarm ด้วยสีเขียวอ่อน ซึ่งสามารถเลือกได้จาก PaintBrush แต่ใน Turbo C ไม่มี ซึ่งทำให้การแสดงผลผิดพลาดไป ซึ่งในกรณีนี้ยังไม่ร้ายแรงเท่าไรนัก (กรณีนี้รวมทั้งการกำหนดตัวแปรชนิด digital และ signal ด้วย) แต่ในกรณีของการสร้างขอบเขตสำหรับการแสดงผลแบบ analog นั้น เราคงต้องพิจารณาการทำงานของการแสดงแบบ analog ก่อน

การแสดงผลแบบ analog นั้น จะรับค่าสัญญาณมาคำนวณ เมื่อกำหนดขอบเขตการแสดงผลแล้วใช้วิธีการ floodfill เมื่อแสดงผลการทำงานของ floodfill นั้นเริ่มจากจุดที่เรากำหนดในเส้นวงปิด แล้วตรวจสอบว่าจุดข้างต้นเป็นสีขอบเขตที่กำหนดหรือไม่ ถ้าไม่ใช่ก็จะทำการเปลี่ยนให้เป็นสีที่เราต้องการ แต่ถ้าเป็นสีขอบเขตก็จะหยุดทำงาน การกำหนดค่าสีขอบเขตนั้นเป็นสิ่งที่สำคัญมาก ถ้าสีขอบเขตที่เรากำหนดโดยใช้ PaintBrush นั้น Turbo C ไม่สามารถรับรู้ได้ (รับรู้จากค่า Attribute ของสี) การ floodfill จะทำงานผิดพลาด ทำให้เราได้ภาพการแสดงผลที่ผิดพลาดไป

ดังนั้นขณะสร้างภาพหลักต้องกำหนดสีของวงปิดที่ต้องการแสดงผลแบบ analog ให้เป็นสีที่ Turbo C รับรู้ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารนี้ทุกครั้งที่มีการนำไปใช้

ผลการทำงาน

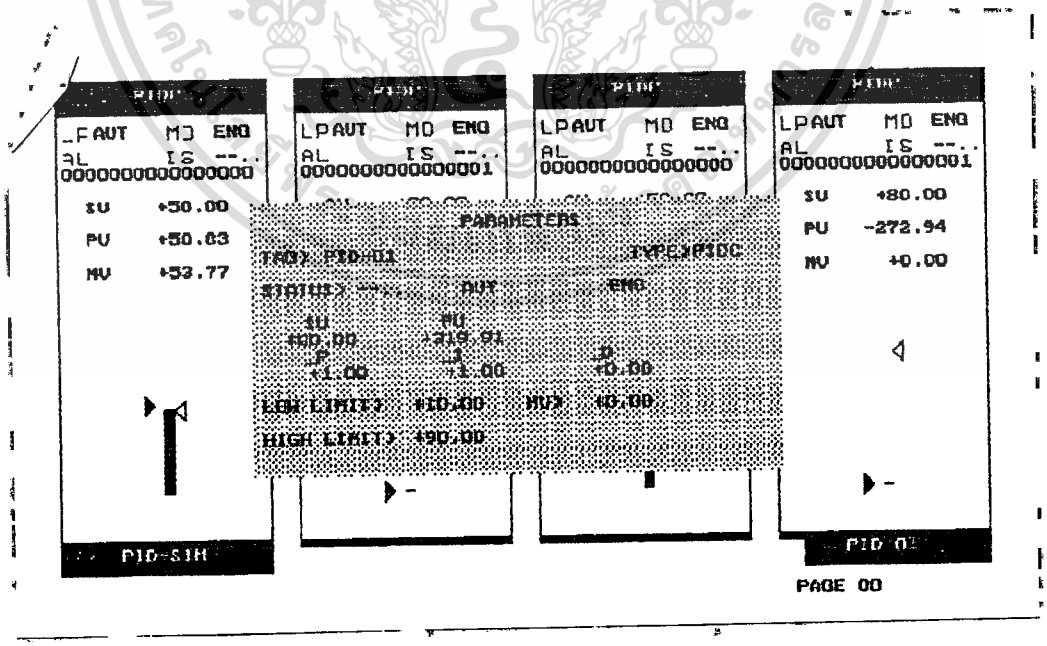
ในส่วน BUILDER ผลของการปรับปรุงการทำงาน โดยมีผลการปรับปรุงดังนี้

1. การใช้งานของเมนูทำให้เกิดความต่อเนื่องในการใช้งานของผู้ใช้

2. การใช้งานของคีย์บอร์ดต่างๆ มีความคล่องตัวมากขึ้น ทำให้การสร้าง Configuration และการติดตั้ง Utility ต่างมีความสะดวกรวดเร็วมากขึ้น อีกทั้งสามารถแก้ไขข้อผิดพลาดต่างๆได้สะดวกรวดเร็วมากขึ้น

3. ทำการรวมเมนูของ builder Graphic เข้าไปในเมนูหลักทำให้ผู้ใช้งานสามารถสร้าง Configuration ต่างๆของ Graphic ได้สะดวกยิ่งขึ้น

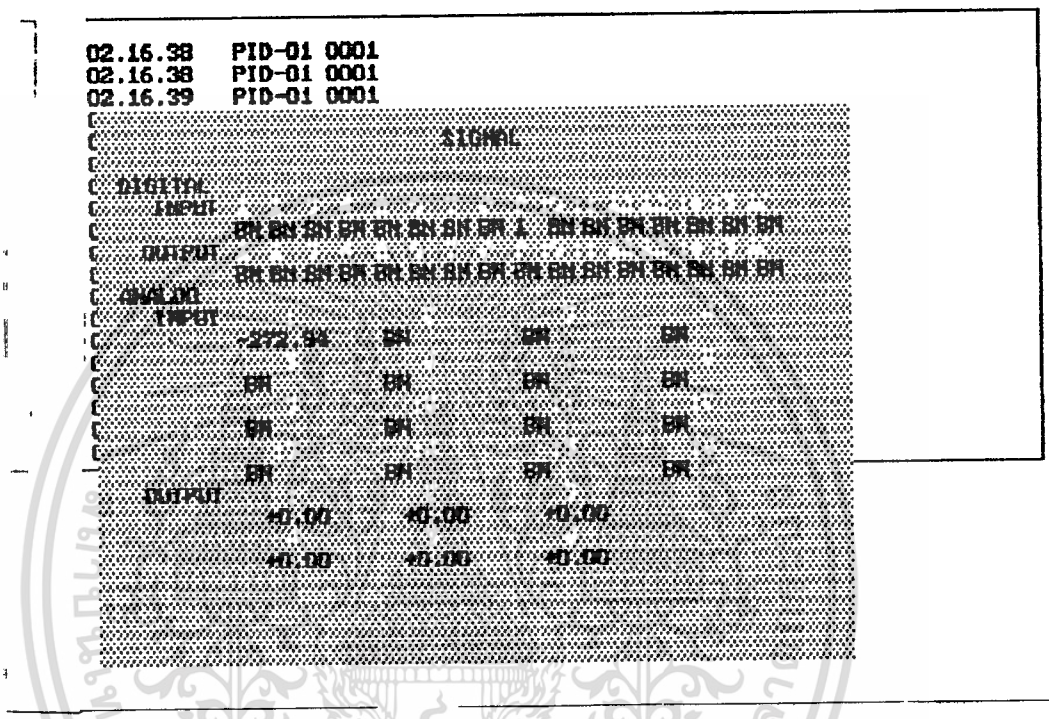
จากการศึกษาโครงสร้างโปรแกรม CTRLLER.EXE เกี่ยวกับโครงสร้างข้อมูลการควบคุมนั้น เราสามารถนำข้อมูลการควบคุมเหล่านั้นซึ่งมีด้วยกันหลายรูปแบบออกมาได้ตามต้องการ ทั้งในรูปการแสดงผลของแบบตัวอักษรและตัวเลข



รูปที่ 16 แสดงการดึงข้อมูลจากบล็อคข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขณะเดียวกันการแก้ไขข้อมูลต่าง ๆ ก็ยังสามารถทำได้เหมือนโปรแกรมเดิม โดยมีวิธีการทำงานที่สะดวกขึ้น

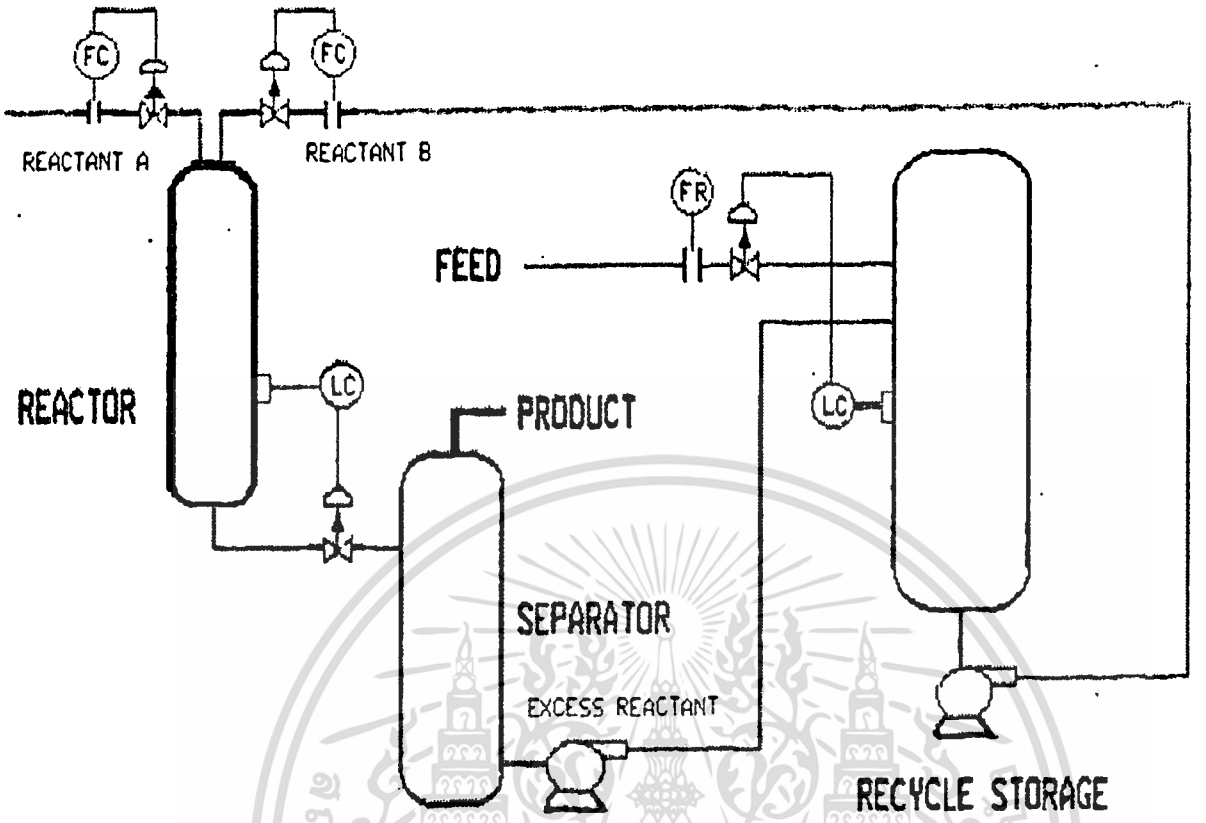


รูปที่ 17 แสดงการดึงข้อมูลจากพอร์ต

สำหรับการสร้างภาพหลักโดยใช้โปรแกรม PC PaintBrush เป็นตัววาดนั้น จากการอ่านภาพจากไฟล์ CPCPIC.PCX แล้วนำขึ้นหน้าจอนั้นได้ผลเป็นที่น่าพอใจ ภาพที่ได้มีความละเอียดชัดเจน ซึ่งภาพจากการอ่านไฟล์นี้สามารถเก็บไว้ในหน่วยความจำของเครื่อง และการนำภาพที่เก็บไว้มาใช้นั้นก็สามารถทำได้โดยการอ่านภาพจากหน่วยความจำมายังส่วนแรมของจอภาพ

ในการเพิ่มทาก์สการทำงานในโปรแกรม CTRLLER.EXE สามารถทำได้อย่างสมบูรณ์ กล่าวคือได้ทำการสร้างทาก์สสำหรับการแสดงผลขึ้นมาอีกหนึ่งทาก์ส ซึ่งทาก์สที่เพิ่มขึ้นนี้สามารถนำไปประยุกต์ใช้ในงานอื่น ๆ ได้อีก เช่น การตรวจสอบสัญญาณร้องขอ (request signal) จากส่วนควบคุมกระบวนการ กรณีที่มีการขยายระบบควบคุมที่ใหญ่ขึ้น เพื่อความปลอดภัยในการทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น และห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารนี้ทุกครั้งที่มีการนำไปใช้



รูปที่ 18 แสดงภาพที่ได้จากการอ่านจากหน่วยความจำ

สำหรับส่วนสุดท้ายคือการนำภาพที่ได้จากหน่วยความจำมาประมวลผลกับข้อมูลการควบคุมเพื่อแสดงในทาก์สใหม่นั้น จากการทดลองยังไม่สามารถทำงานได้ จากการทดสอบ การนำภาพจากหน่วยความจำมาแสดงสามารถทำได้เมื่อไม่ได้อยู่ในการทำงานแบบมัลติทาก์สกิง สมมุติฐานของปัญหาจึงอาจเกิดจากเรื่องหน่วยความจำของทาก์สต่าง ๆ และของภาพที่อาจซ้ำซ้อนกัน ซึ่งยังไม่สามารถสรุปได้

แต่ลักษณะการทำงานนั้นสามารถจำลองให้ดูได้ ซึ่งอัลกอริธึมของโปรแกรมจำลองจะเหมือนกับกรณีที่ใช้งานจริงในโปรแกรม CTRLLER.EXE ซึ่งผลที่ได้จากการจำลองนั้นสามารถทำงานได้ดี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สรุปผล

สรุปผลการทำงานของซอฟท์แวร์

จากการทำงานของโปรแกรมในส่วนการแสดงผลติดต่อกับผู้ใช้งานถือได้ว่ามีความสำเร็จในระดับหนึ่ง กล่าวคือสามารถทำงานร่วมกับตัวโปรแกรมเดิมที่ทำงานในส่วนควบคุมได้ ซึ่งก็คือการดึงข้อมูลจากโปรแกรมเดิมมาจัดการเพื่อการแสดงผล และในส่วนการสร้างข้อกำหนดการควบคุมสามารถนำโปรแกรมใหม่เข้าแทรกเป็นส่วนหนึ่งของการทำงานเดิม ดังนั้นเราคงพอสรุปได้ว่าปัญหาใหญ่ในการสร้างโปรแกรมส่วนการแสดงผลนี้ก็คือการดึงเอาข้อมูลที่ตัวควบคุมใช้งานอยู่มาแสดงผลตามรูปแบบต่างๆ และการนำการแสดงผลที่กำหนดขึ้นรวมกับส่วนควบคุม ซึ่งเราสามารถบรรลุในส่วนนี้ได้

ในส่วนต่อไปที่สำคัญ คือรูปแบบของการแสดงผลที่ทำให้ผู้ใช้สามารถใช้งานได้สะดวกขึ้น จากโปรแกรมที่สร้างขึ้นการแสดงผลในโหมดตัวอักษรเป็นการแสดงค่าข้อมูลโดยรวมซึ่งข้อมูลที่ได้ไม่ได้มีจุดมุ่งหมายที่จะแสดงผลแบบ real time แต่ต้องการแสดงภาพรวมของข้อมูลมากกว่า

สำหรับการกำหนดการแสดงผลในโหมดกราฟิกซึ่งเป็นอีกการทำงานหนึ่งในโปรแกรม BUILDER ในส่วนนี้ถ้ามองเฉพาะหลักการทำงานถือว่าใช้ได้ แต่ในด้านการใช้งานนั้นค่อนข้างยุ่งยาก ผู้ใช้งานต้องทราบวิธีการใช้งานอย่างลึกซึ้งจึงสามารถใช้งานได้ดี ซึ่งคงต้องได้รับการพัฒนาต่อไป

และสำหรับการนำภาพหลักไปประมวลผลในส่วนการควบคุมซึ่งไม่สำเร็จนั้น ถึงแม้ว่าจะไม่สามารถระบุที่มาของปัญหาได้ชัดเจน แต่จากการจำลองการทำงานแสดงให้เห็นว่าการนำภาพหลักมาประมวลผลกับข้อมูลจากการควบคุมนั้นสามารถทำได้ดีพอสมควร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การพัฒนาขั้นต่อไป

การพัฒนาขั้นต่อไปสำหรับโปรแกรมลักษณะนี้ คือ การขยายระบบควบคุมให้แยกส่วนการทำงานออกจากกัน โดยแบ่งเป็น ส่วนที่ควบคุมกระบวนการ และส่วนควบคุมหลัก โดยในส่วนควบคุมกระบวนการนั้น ประกอบด้วยทาก์ส (task)การควบคุมทาส์กการสื่อสาร และระบบรักษาความปลอดภัยกรณีขาดการติดต่อกับส่วนควบคุมหลัก (FAIL SAFE) สำหรับที่ส่วนการควบคุมใหญ่นั้นจะมี ทาก์สการแสดงผล การควบคุมการทำงาน (supervisory) การสื่อสาร ซึ่งทาก์สนี้แตกต่างจากทาก์สในส่วนควบคุมกระบวนการ ซึ่งถ้าเรานำโครงสร้างเดิมของโปรแกรมนี้เป็นหลัก เราสามารถแก้ไขโปรแกรมนี้ให้เป็นโปรแกรมของส่วนควบคุมหลักได้อย่างง่าย ๆ คือ

การแก้ไขฟังก์ชั่น void usr_interrupt (void) ที่โปรแกรม trendrec.c ในส่วนฟังก์ชั่น InstCalc() โดยเปลี่ยนเป็นฟังก์ชั่นการสื่อสารแทน และขยายการทำงานของโปรแกรม MANAGER.C ให้ทำงานควบคุมได้กว้างขึ้น

ซึ่งเมื่อถึงจุดนั้นแล้วเรายังสามารถพัฒนาส่วนควบคุมหลักให้มีการทำงานแบบ REDUNDANCY ได้

สำหรับผู้ที่ต้องการพัฒนาโปรแกรมนี้ต่อไป ควรระวังเรื่องความคอมแพทติบิลิตี้ของโปรแกรม เพราะที่ผ่านมาการพัฒนาโปรแกรมนี้ทั้งสองปีการศึกษาทำอยู่บนเครื่องคอมพิวเตอร์เครื่องเดียวกัน ซึ่งลักษณะของเครื่องอาจผิดจากเครื่องอื่น ๆ โดยเฉพาะประเภท multitasking ดังนั้นการพัฒนาขั้นต่อไปอาจทำโดยการดัดแปลงโครงสร้างโปรแกรมเดิมแล้วนำไปใช้กับเครื่องอื่น ๆ และเพื่อความสะดวกเครื่องคอมพิวเตอร์ที่ใช้ควรมีความสามารถในการขยายหน่วยความจำแรม (RAM) ได้เกินกว่า 1 MB เพื่อความสะดวกในการพัฒนาโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้ได้สำเร็จลุล่วงไปได้โดยความช่วยเหลือ และให้คำแนะนำที่ดี รวมไปถึงการช่วยแก้ปัญหาต่างๆ ของ อาจารย์ สุเธียร เกียรติสุนทร อาจารย์ที่ปรึกษา

ขอขอบคุณ อาจารย์ จงกล งามวิวิทย์ อาจารย์ เกียรติวรรณ ทรงสัตย์ ที่ให้ความสะดวกในการใช้เครื่องคอมพิวเตอร์

ขอขอบคุณ นายชัยเดช วรรณผ่องใส และ นายธรณิน แก่นสันติสุขมงคล ที่ริเริ่มทำโครงงานนี้ขึ้นมา เพื่อให้ผู้เขียนได้ทำการสานต่อจนสำเร็จไปด้วยดี

ขอขอบคุณ นายวิเชียร อึ้งสกุลรัตน์ ที่ช่วยสละเวลาอันมีค่ามาช่วยพิมพ์วิทยานิพนธ์ฉบับนี้

ขอขอบพระคุณ บิดามารดา ที่คอยสนับสนุนการเงิน และให้กำลังใจต่อผู้เขียนเสมอมาจนกระทั่งสำเร็จการศึกษา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

[1] Herbert Schildt , Turbo C/C++ The Complete Reference, Osborne McGraw-Hill ,1990

[2] Herbert Schildt , Born to Code in C , Osborne McGraw-Hill, 1990

[3] Michael P. Lukas , Distributed Control System , Van Norstrand ReinHold Company Inc. ,1986

[4] Nelson Johnson , Advance Graphic in C : Programming and Techniques , Osborne McGraw-Hill ,1991

[5] ชัยยุทธ ตันประทุมวงษ์ , แกะรอย PCX ด้วยเทอร์โบซี ไมโครคอมพิวเตอร์ ฉบับที่ 78 , มกราคม 2535

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้