



ระบบควบคุมแบบกระจายส่วน

Distributed Control System : DCS

โดย

นางสาว นัตริลดา ทองชั้น 37014074

นาย ภิญโญ แซ่ตั้ง 37014320

อาจารย์ที่ปรึกษา

อาจารย์ สุเชียร เกียรติสุนทร

วัน เดือน ปี.....	11. ๑๒. 2541
เลขทะเบียน.....	038871
เลขเรียกหนังสือ.....	๓๐๓๕ ๓๓ ๖

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิศวกรรมระบบควบคุม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2540



038871

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ทางการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาโทปีการศึกษา 2540

ภาควิชาวิศวกรรมระบบควบคุม

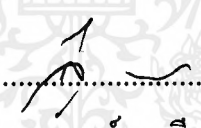
คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบควบคุมแบบกระจายส่วน

Distributed Control System : DCS

จัดทำโดย

1. นางสาว ฉัตรลดา ทองชั้น
2. นาย ภิญโญ แซ่ตั้ง


..... อาจารย์ที่ปรึกษา
(อาจารย์ สุเชียร เกียรติสุนทร)

ระบบควบคุมแบบกระจายส่วน
Distributed Control System : DCS

โดย
นางสาว ฉัตรลดา ทองชั้น
นาย ภิญโญ แซ่ตั้ง

อาจารย์ที่ปรึกษา

อาจารย์ สุเชียร เกียรติสุนทร

บทคัดย่อ

ปริญญาานิพนธ์ฉบับนี้ ได้ศึกษาถึงระบบการควบคุมแบบกระจายส่วน ซึ่งเป็นการใช้โครงข่ายในการควบคุมระบบหรือโรงงานต่าง ๆ ในระยะไกล ระบบที่ถูกทำการควบคุมอาจจะเป็นระบบอย่างง่ายที่ถูกควบคุมโดยการใช้การควบคุมแบบย้อนกลับ หรืออาจจะเป็นระบบที่ซับซ้อนที่ถูกควบคุมโดยการใช้ระบบควบคุมแบบต่อเนื่อง , ระบบควบคุมแบบป้อนไปข้างหน้า - ป้อนกลับ หรือระบบควบคุมแบบติด-ดับ เป็นต้น โครงการงานชิ้นนี้เป็นความพยายามที่จะสร้างระบบควบคุมขึ้นมาเพื่อเลียนแบบระบบควบคุมที่ใช้งานจริงในวงการอุตสาหกรรม ความจริงแล้วโปรแกรมการควบคุมเริ่มต้นของโครงการงานชิ้นนี้ได้ถูกสร้างขึ้นโดยอาจารย์ในภาควิชาวิศวกรรมระบบควบคุม แต่เป็นระบบควบคุมที่สามารถควบคุมได้เฉพาะระบบที่ติดต่อกับเครื่องคอมพิวเตอร์ที่ใช้ควบคุมเครื่องเดียวเท่านั้น ยังไม่มีส่วนของการควบคุมที่สามารถควบคุมได้เป็นเครือข่าย ดังนั้นงานของคณะผู้จัดทำคือการสร้าง โปรแกรมในส่วนของการควบคุมแบบเครือข่าย แล้วนำไปเพิ่มเข้าไปในโปรแกรมเริ่มต้นเดิม นอกจากนี้ทางคณะผู้จัดทำยังได้ทำการสร้างโปรแกรมขึ้นมาช่วยในการสนับสนุนการเข้าถึงในส่วนต่าง ๆ ของโปรแกรมโดยการใช้เมาส์ ซึ่งทำให้เกิดความสะดวกในการใช้โปรแกรมมากกว่าโปรแกรมเดิม ซึ่งต้องใช้คีย์บอร์ดในการเข้าถึงเพียงอย่างเดียว

ABSTRACT

The Distributed Control System is the network of controlling the process or plant from the distance. The Process can be the simple one operated with the feedback control or a complex one by mean of using cascade control , FF-FB control, On/Off control, etc. This project is a simulated version of the one that is used in the factory system. The program which is written by the teacher in Control Department can control only the process

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
แม้ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

that connect to the computer unit itself. It does not have a supporting for the controlling by network. So creating the program of the network part and adding it to the old version is our task. Furthermore we also create the program about accessing the parts of the control program by using mouse. This can make the accessing more comfortable than using the keyboard.



กิตติกรรมประกาศ

ปริญญาบัตรฉบับนี้สำเร็จลุล่วงไปได้ด้วยดี เนื่องจากได้รับคำแนะนำ ความรู้ และ ความช่วยเหลือจากอาจารย์ สุเชียร เกียรติสุนทร อาจารย์ที่ปรึกษา จึงขอขอบพระคุณมา ณ ที่นี้ ด้วย

ขอขอบพระคุณคุณพ่อ คุณแม่ และ พี่น้อง ที่ให้กำลังใจมาโดยตลอด และขอขอบคุณเพื่อน ๆ ที่ให้ความช่วยเหลือ กำลังใจ และความเป็นความเป็นเพื่อนที่ดี ตลอดมา

คณะผู้จัดทำ

9/4/41



สารบัญ

เรื่อง	หน้า
บทที่ 1 บทนำ	1
บทที่ 2 ทฤษฎีและหลักการ	8
- ทฤษฎี	8
- หลักการ	13
บทที่ 3 ส่วนประกอบของงานที่ทำและการเปรียบเทียบ	14
- ส่วนของการเขียนโปรแกรมในการสนับสนุนการติดต่อสื่อสารภายในระบบ	14
- ส่วนของการเพิ่มโปรแกรมสนับสนุนการติดต่อสื่อสารเข้าไปในโปรแกรมหลัก	21
- ส่วนของการสร้างโปรแกรมในส่วนของการสนับสนุนการใช้เมาส์	21
- ข้อแตกต่างระหว่างระบบที่เพิ่มเติมส่วนสนับสนุนกับระบบเดิม	24
- ข้อแตกต่างระหว่างระบบควบคุม DCS ที่ถูกพัฒนาแล้วกับระบบควบคุม DCS ที่ใช้จริงในวงการอุตสาหกรรม	25
บทที่ 4 ลักษณะโดยรวมของระบบ	27
- Human Interface	27
- หน้าจอ Operating System	29
- หน้าจอ Control Unit	57
บทที่ 5 ขั้นตอนการทำงานและแบบฝึกหัดการใช้โปรแกรม	59
- ขั้นที่ 1 การรีเซทระบบ	59
- ขั้นที่ 2 การสร้างบล็อกและขั้นที่ 3 การเชื่อมบล็อก	60
- ขั้นที่ 4 การวาดรูป Process	63
- ขั้นที่ 5 การเซทหน้าจอแสดงผล	68
บทที่ 6 สรุปและวิจารณ์	71

สารบัญตาราง

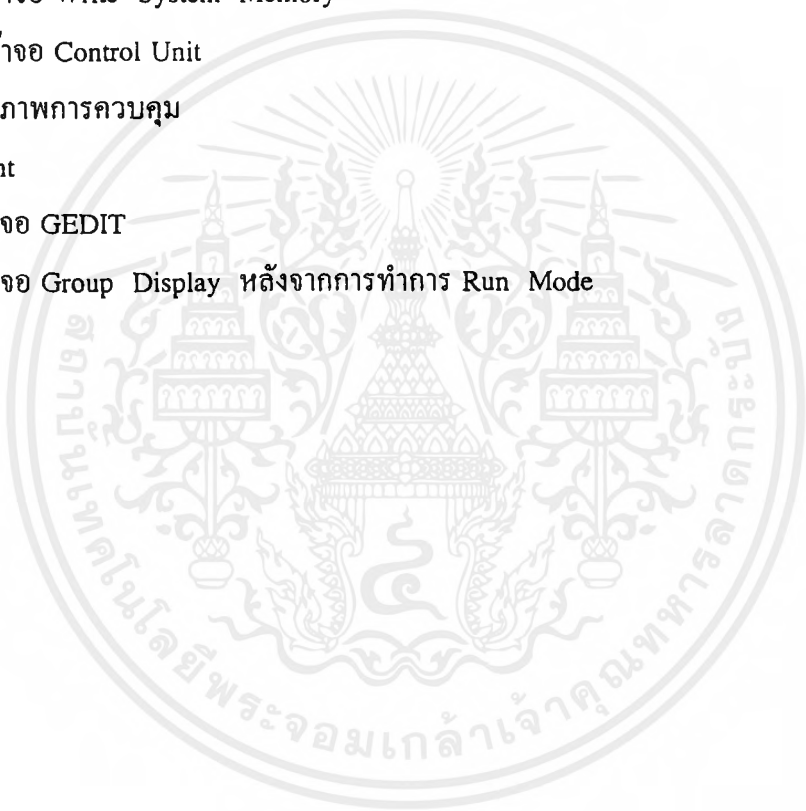
	หน้า
ตารางที่ 1.1 แสดงการเปรียบเทียบระบบควบคุมแบบต่างๆ	7
ตารางที่ 3.1 แสดงรูปแบบการจัดลำดับความสามารถในการส่งข้อมูลใน LAN	18



สารบัญญภาพ

รูปที่	หน้า
1.1 แสดงบล็อกไดอะแกรมการควบคุมแบบดิจิทัลโดยตรง	2
1.2 แสดงโครงสร้างระบบควบคุมแบบผสม	3
1.3 แสดงโครงสร้างระบบควบคุมแบบศูนย์รวม	4
1.4 แสดงโครงสร้างระบบควบคุมแบบกระจาย	4
2.1 แสดงวิวัฒนาการของระบบควบคุม	8
2.2 ระบบควบคุมแบบป้อนกลับ	9
2.3 ถึงผสมถ่ายเทความร้อน	12
3.1 Token Bus	19
3.2 แสดงลำดับขั้นตอนในการทำงาน	20
3.3 แสดงระบบควบคุม DCS ที่เป็นโครงการนของคณะผู้จัดทำ	25
3.4 แสดงระบบควบคุม DCS ที่ใช้จริงในวงการอุตสาหกรรม	25
4.1 แสดงตัวอย่างหน้าจอที่มีการใช้เมาส์	28
4.2 แสดงหน้าจอเมนูหลัก	29
4.3 แสดงหน้าจอ Alarm List	30
4.4 แสดงหน้าจอ Control Block Diagram	31
4.5 แสดงหน้าจอ Control Block Faceplate	33
4.6 แสดงหน้าจอ Clear System Memory	34
4.7 แสดงหน้าจอ Digital Block Diagram	35
4.8 แสดงหน้าจอ Digital Block Faceplate	37
4.9 แสดงหน้าจอ Edit System Memory	38
4.10 แสดงหน้าจอ Exit	39
4.11 แสดงหน้าจอ Graphic Display	40
4.12 แสดงหน้าจอ Graphic Display Generator	41
4.13 แสดงหน้าจอ Group Display	42
4.14 แสดงหน้าจอ Input Block Configuration	43
4.15 แสดงหน้าจอ Initial Screen Memory	44
4.16 แสดงหน้าจอ Network System	45

4.17	แสดงหน้าจอ Output Block Configuration	46
4.18	แสดงหน้าจอ Run Mode	47
4.19	แสดงหน้าจอ Read System Memory	48
4.20	แสดงหน้าจอ System Configuration	49
4.21	แสดงหน้าจอ Sequential Control Monitor	52
4.22	แสดงหน้าจอ Sequential Control Program	53
4.23	แสดงหน้าจอ Standby Mode	54
4.24	แสดงหน้าจอ Trend Display	55
4.25	แสดงหน้าจอ Write System Memory	56
4.26	แสดงหน้าจอ Control Unit	57
5.1	แสดงแผนภาพการควบคุม	59
5.2	แสดง Plant	63
5.3	แสดงหน้าจอ GEDIT	64
5.4	แสดงหน้าจอ Group Display หลังจากการทำการ Run Mode	70



บทที่ 1 บทนำ

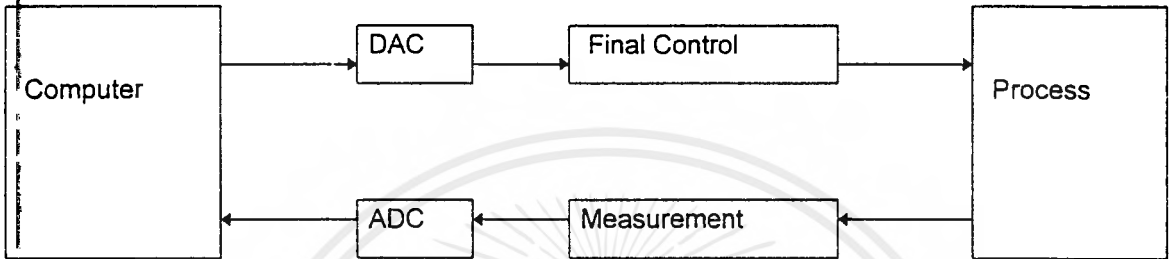
หลักการของระบบควบคุมโดยใช้คอมพิวเตอร์ไม่ได้เป็นของใหม่เท่าใดนักตามจริงแล้วก็คือ การนำข้อมูลจากอุปกรณ์ควบคุมและอุปกรณ์วัดค่าต่างๆในกระบวนการทำงาน (Process) มาประมวลผล และส่งสัญญาณควบคุมออกไป โดยเปลี่ยนการควบคุมโดยใช้มนุษย์มาเป็น การควบคุมแบบระบบอัตโนมัติ และเป็นการทำงานแบบอนาล็อกมาเป็นแบบดิจิทัล ซึ่งในระยะเริ่มต้นของการควบคุม การควบคุมกระบวนการจะเป็นแบบอนาล็อกทั้งหมดโดยสัญญาณการควบคุมเป็นความดันอากาศ (Air Pressure) หรือกระแสไฟฟ้า โดยเรียกระบบการควบคุมแบบนี้ว่า ระบบควบคุมแบบนิวเมติก (Pneumatic Process Control System) และระบบควบคุมแบบอิเล็กทรอนิกส์ (Electronic Process Control System) ตามลำดับ ตัวอย่างที่เห็นได้ชัดเจนก็คือเครื่องโปรแกรมเมเบิลลอจิกคอนโทรลเลอร์ (Programmable Logic Controller) ซึ่งมีการนำมาใช้งานอย่างชัดเจนตั้งแต่ปี ค.ศ. 1970

ต่อมาเมื่อระบบทางดิจิทัลเริ่มมีบทบาทเข้ามาแทนที่ระบบอนาล็อกมากขึ้นในรูปการควบคุมจึงมีการนำระบบดิจิทัลเข้ามาช่วยในการควบคุม โดยในระยะแรกเริ่มนั้นได้มีการนำระบบดิจิทัลเข้ามาช่วยในการบันทึกข้อมูล ลักษณะค่าค่าล็อกกิ้ง (Data Logging) ซึ่งมีส่วนช่วยในการวิเคราะห์ผลการควบคุมจากข้อมูลหรือการควบคุมที่ผ่านมา จากการบันทึกด้วยเครื่องบันทึก (Recorder) หรือที่บันทึกด้วยชาร์ต (Chart) ต่างๆ โดยรับสัญญาณอนาล็อกจากอุปกรณ์แล้วแปลงเป็นสัญญาณดิจิทัลโดยตัวแปลงจากสัญญาณอนาล็อกเป็นสัญญาณดิจิทัลคือ เอดีซี (ADC) ระบบควบคุมโดยใช้คอมพิวเตอร์เริ่มมีความจำเป็นต่อการใช้งานมากขึ้นเรื่อยๆ เนื่องจากระบบควบคุมยุคใหม่ มีความสลับซับซ้อนมากขึ้นและต้องการความแม่นยำในการควบคุมมากขึ้น งานแรกที่มีการนำคอมพิวเตอร์มาใช้งานได้แก่ งานแสดงผลการและดูแลการควบคุม (Supervisory Control) และหลังจากนั้นได้มีการนำคอมพิวเตอร์มาควบคุมการปรับจุดเป้าหมาย (Set Point) เพื่อเพิ่มประสิทธิภาพในการควบคุม ซึ่งสัญญาณอ้างอิงเหล่านี้จะถูกแปลง แล้วส่งไปยังตัวควบคุมชนิดอนาล็อก

ในที่สุดเมื่อระบบดิจิทัล ได้เข้ามาแทนที่ระบบอนาล็อกเกือบทั้งหมด ทำให้เกิดการนำคอมพิวเตอร์มาใช้เป็นตัวควบคุมโดยตรง ซึ่งเรารู้จักกันในนามของ Direct Digital Control (DDC) หรือ Computer Process Control ซึ่งระบบ DDC นี้ได้มีการเริ่มนำใช้งานเป็นครั้งแรกในโรงงานปิโตรเคมี เมื่อปี ค.ศ. 1963 แต่เพื่อความปลอดภัยในช่วงแรกของการใช้งาน ได้มีการนำระบบควบคุมแบบอนาล็อกมาช่วยใช้ร่วมด้วย เพื่อสำรองกรณีระบบคอมพิวเตอร์เกิดขัดข้อง ข้อดีของระบบควบคุมโดยใช้คอมพิวเตอร์นี้มีหลายประการ เช่น ในการปรับค่าพารามิเตอร์ต่างๆ จะไม่เกิดการครีฟ (Drift) ขึ้น ความสามารถในการสร้างการควบคุมที่สลับซับซ้อน และความสามารถในการปรับค่าพารามิเตอร์ตามสถานะที่เปลี่ยนแปลงได้อย่างเหมาะสม (Optimum Control) ซึ่งแผนผัง

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อใช้ประโยชน์ในการศึกษาวิจัยเท่านั้น ขอสงวนสิทธิ์ในการค้า
ไว้ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แสดงลักษณะของระบบควบคุมแบบ DDC สามารถแสดงได้ดังรูปที่ 1.1 มีส่วนควบคุมเป็นคอมพิวเตอร์ ส่วนตัวแปลงจากอนาล็อกเป็นดิจิทัล (Analog to Digital -Converter : ADC) และตัวแปลงจากดิจิทัลเป็นอนาล็อก (Digital to Analog Converter : DAC) เป็นตัวแปลงสัญญาณเพื่อติดต่อกับกระบวนการ สำหรับการทำงานของส่วนควบคุม ก็จะเป็นการกำหนดจากโปรแกรมหรือซอฟต์แวร์

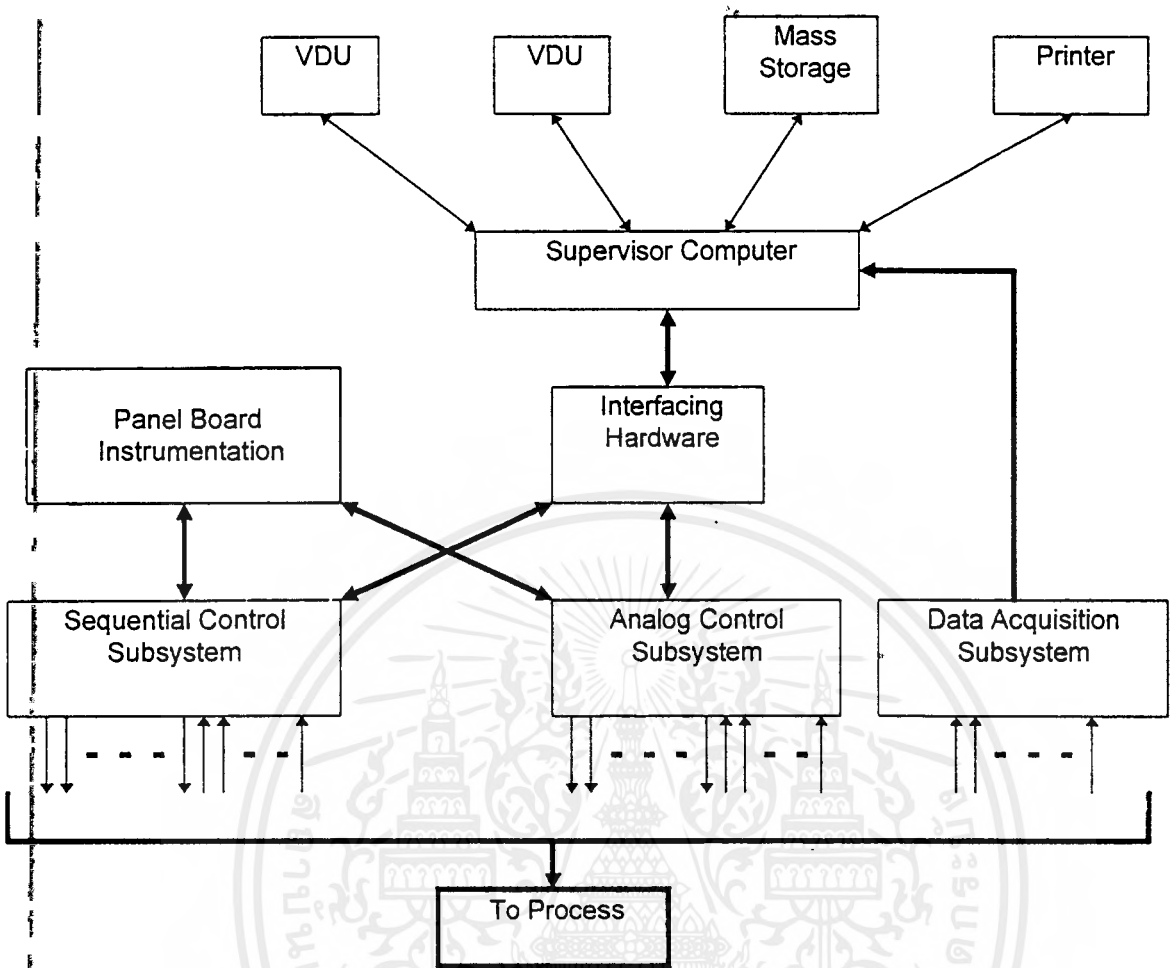


รูปที่ 1.1 แสดงบล็อกไดอะแกรมการควบคุมแบบดิจิทัลโดยตรง

จากการพัฒนาระบบควบคุมจนถึงปัจจุบัน เราสามารถแบ่งโครงสร้างระบบควบคุมออกเป็น 3 ลักษณะ

1. โครงสร้างแบบไฮบริดจ์ (Hybrid Architecture)

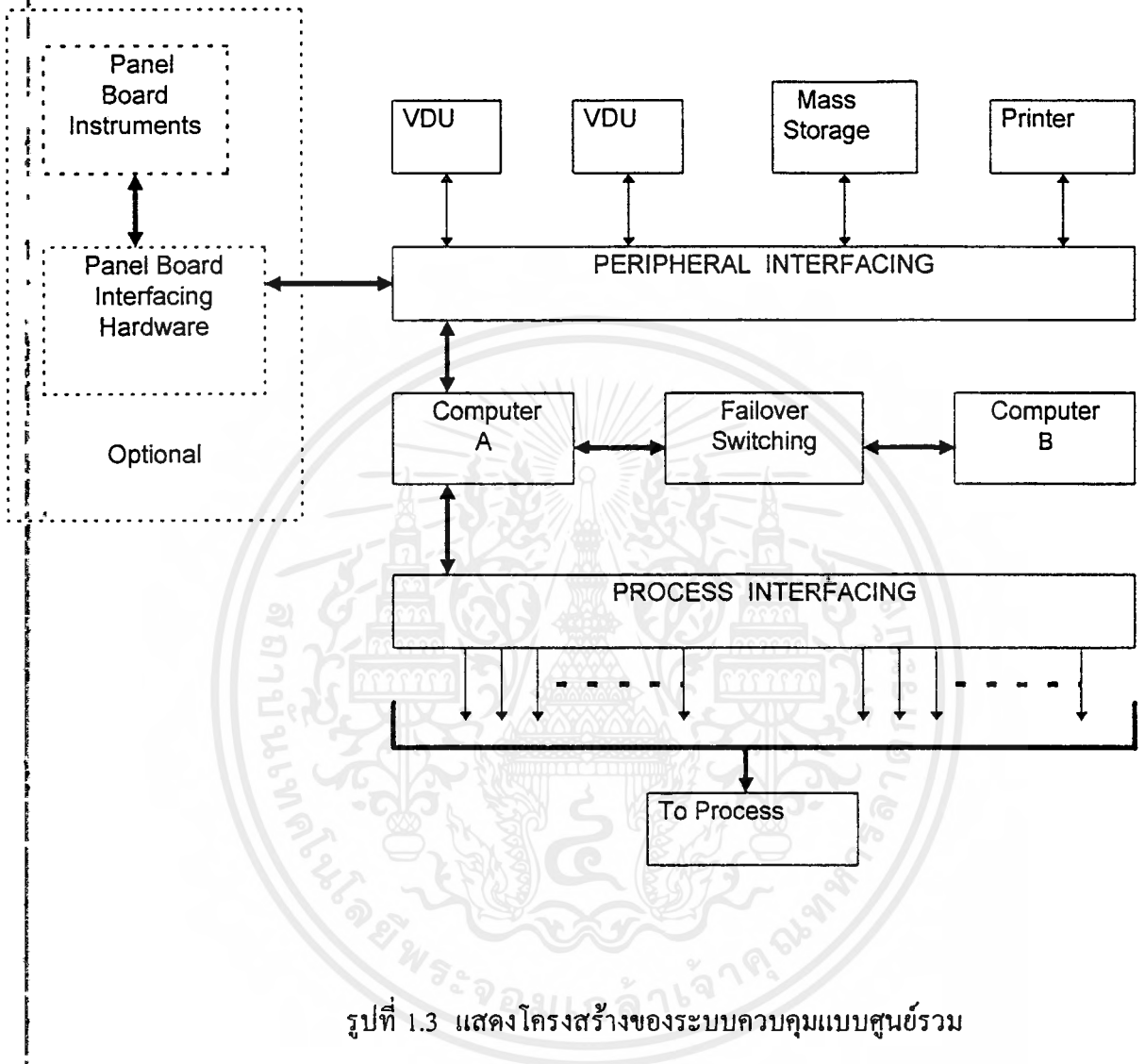
โครงสร้างแบบผสม (Hybrid System) เป็นผลรวมของการใช้งานของฮาร์ดแวร์การควบคุมแบบดิสครีต (Discrete Control Hardware) และฮาร์ดแวร์ของคอมพิวเตอร์ (Computer Hardware) ที่ตำแหน่งศูนย์กลางของการควบคุม เมื่อใช้งานเกี่ยวกับฟังก์ชันที่เพิ่มขึ้นมาตามความต้องการในโครงสร้างแบบนี้ การควบคุมแบบท้องถิ่น (Local Control) ในหน่วยของกระบวนการควบคุมแบบซีแควนเชียล (Sequential Discrete Analog) หรือ PLC การวัดของแผงควบคุมติดต่อกับควบคุมเหล่านี้ โดยใช้สำหรับผู้ควบคุมและติดตั้งอยู่ในห้องควบคุมกลาง (Central Control Room Area) นอกจากนี้ยังมีการนำเอาซูเปอร์ไวเซอร์คอมพิวเตอร์ (Supervisor Computer) และระบบการรวบรวมข้อมูลมาช่วยในการจัดการต่างๆ ของขบวนการ รวมไปถึงการควบคุมขบวนการให้เหมาะสม การเตือนภัย การส่งข้อมูล การเก็บข้อมูล และการนำข้อมูลเหล่านี้มาใช้งาน ในส่วนของการติดต่อกับผู้ควบคุมจะประกอบด้วยอุปกรณ์แสดงผลตั้งแต่ 1 ตัวขึ้นไป ซึ่งระบบการควบคุมที่มีโครงสร้างแบบไฮบริดจ์นี้ สามารถแสดงได้ดังรูปที่ 1.2



รูปที่ 1.2 แสดงโครงสร้างระบบควบคุมแบบผสม

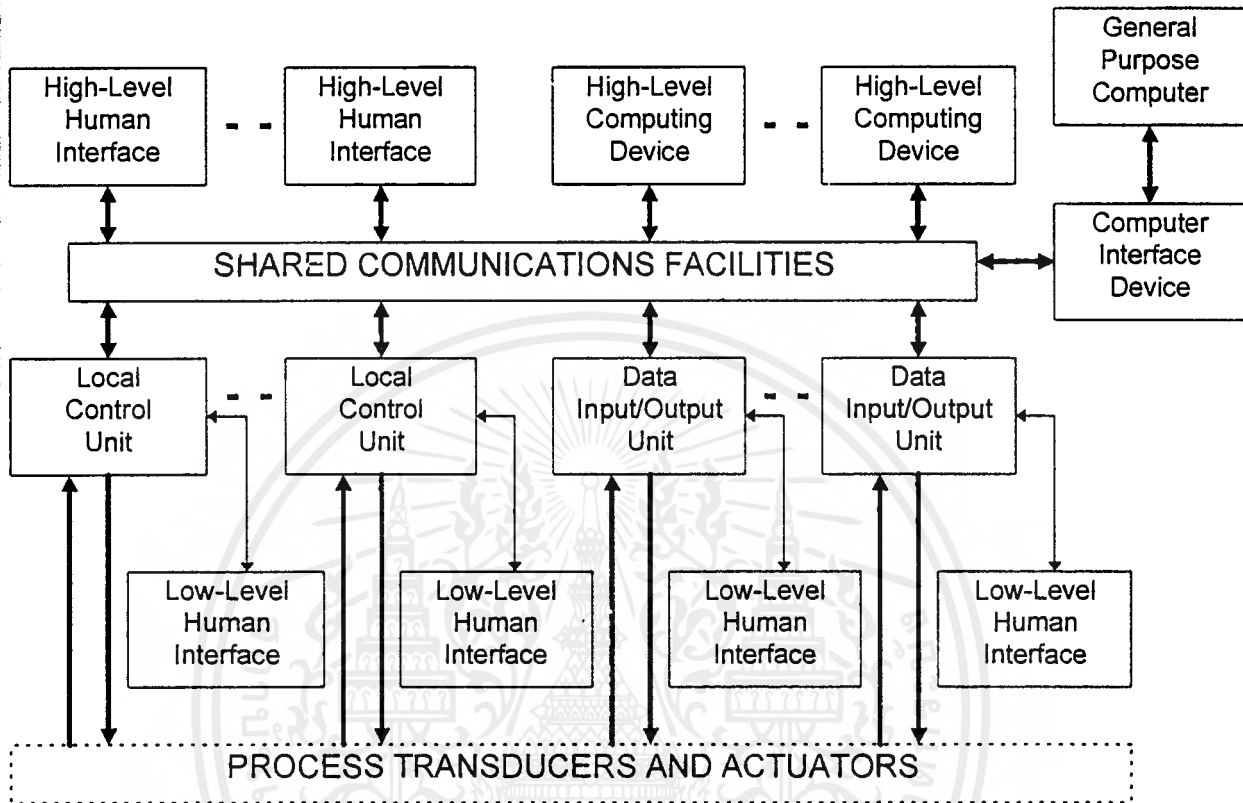
2. Central Computer System

ส่วนโครงสร้างชนิดนี้สามารถแสดงให้เห็นได้ดังรูปที่ 1.2 ซึ่งเป็นระบบที่เพิ่มเติมอุปกรณ์ที่มีประสิทธิภาพสูงเข้าไปในห้องควบคุม โดยทั่วไปจะต้องมีการนำเอาคอมพิวเตอร์สำรอง (Redundancy) มาเพิ่มเข้าไปในกรณีที่คอมพิวเตอร์หลักไม่สามารถทำงานได้ตามปกติ ส่วนการติดต่อใช้งานของผู้ควบคุม สามารถทำได้ผ่านอุปกรณ์รับและแสดงผลเหมือนกับโครงสร้างการควบคุมแบบผสมที่กล่าวมาแล้วข้างต้น ส่วนทางเลือกอื่นๆ ได้แก่ การใช้งานคอมพิวเตอร์ร่วมกับแผงวัดของการควบคุม (Panel Instrumentation) ทำให้ผู้ควบคุมทำการติดต่อในระดับแรกได้เหมือนตามปกติ



รูปที่ 1.3 แสดงโครงสร้างของระบบควบคุมแบบศูนย์รวม

3. Distributed Control System



รูปที่ 1.4 แสดงโครงสร้างของระบบควบคุมแบบกระจาย

จากรูปที่ 1.4 ซึ่งแสดงส่วนโครงสร้างของระบบควบคุมแบบกระจายไว้ จะเห็นได้ว่าโครงสร้างประกอบด้วยส่วนประกอบที่มากกว่า โครงสร้างการควบคุมแบบอื่นๆ ซึ่งอธิบายโดยสรุปได้ดังนี้

- Local Control Unit (LCU) เป็นหน่วยงานที่เล็กที่สุดในระบบซึ่งมีความสามารถในการควบคุมแบบปิด (Closed - Loop Control) ได้ ตัว LCU จะเป็นตัวรับและส่งสัญญาณเข้ากับกระบวนการผลิต (Process) โดยตรง
- Low - Level Human Interface (LLHI) เป็นอุปกรณ์ให้ผู้ควบคุมสามารถติดต่อกับ LCU ในการปรับเปลี่ยนค่าพารามิเตอร์ต่างๆ ได้โดยตรง
- Data Input / Output Unit (DI/OU) เป็นอุปกรณ์ในการรับส่งสัญญาณจากกระบวนการผลิตเพื่อนำไปใช้ในงานอื่นๆ ต่อไป

- High - Level Human Interface (HLHI) เป็นชุดอุปกรณ์ที่ทำหน้าที่คล้ายกับ LLHI แต่ได้เพิ่มประสิทธิภาพการทำงาน และความสะดวกในการใช้งานให้มากขึ้น นอกจากนี้ยังสามารถติดต่อกับอุปกรณ์ควบคุมตัวอื่นๆ โดยผ่านระบบการสื่อสารข้อมูลร่วมได้

- Computer Interface Device (CID) เป็นอุปกรณ์สำหรับคอมพิวเตอร์นอกระบบที่จะเข้ามาติดต่อกับระบบควบคุมนี้

- ระบบสื่อสารร่วมเป็นอุปกรณ์ทั้งฮาร์ดแวร์และซอฟต์แวร์ที่ทำหน้าที่ในการจัดการจัดสรรข้อมูล ระหว่างหน่วยงานทำงานต่างๆ ภายในระบบให้ได้รับข้อมูลที่หน่วยงานทำงานนั้น เกี่ยวข้องไปปฏิบัติงาน



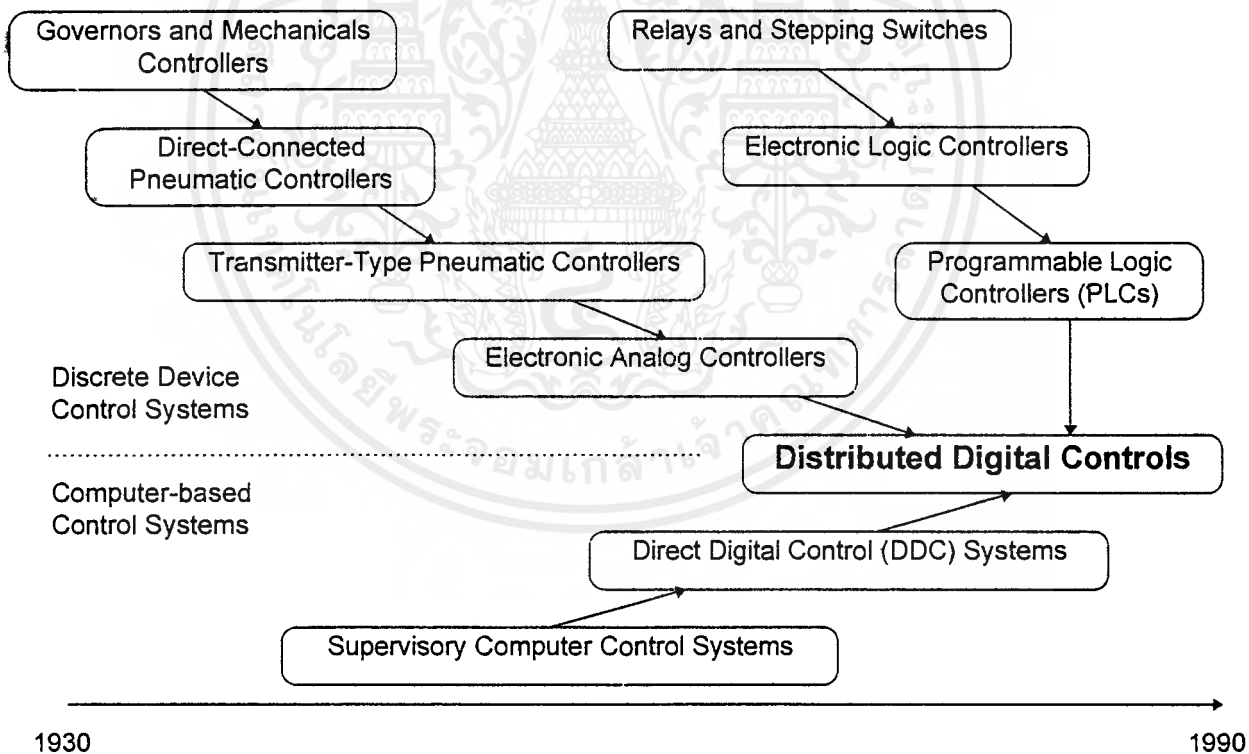
โครงสร้าง	Hybrid	แบบศูนย์รวม	แบบกระจาย
1. ขอบเขตการใช้งาน	ดีขึ้นกับรูปแบบโครงสร้าง	ไม่ดีเนื่องจากการจำกัดขอบเขตของระบบ	ดีขึ้นอยู่กับรูปแบบโครงสร้าง
2. ความสามารถในการควบคุม	ถูกจำกัดโดย เครื่องควบคุมอนาล็อกและซีเควนเชียล	ความสามารถทางด้านการควบคุมทางดิจิทัลเพิ่มเติมที่	ความสามารถทางด้านการควบคุมทางดิจิทัลเพิ่มเติมที่
3. ความสามารถในการติดต่อกับผู้ใช้	ถูกจำกัดโดยเครื่องมือที่ใช้ป้อนข้อมูล	มีเครื่องมือดิจิทัลเตรียมการพัฒนาที่สำคัญสำหรับระบบขนาดใหญ่	มีเครื่องมือดิจิทัลเตรียมการพัฒนาสำหรับระบบที่เต็มพิกัด
4. การอินทิเกรตของฟังก์ชันระบบ	ต่ำขึ้นกับความหลากหลายของผลิตภัณฑ์	ทุกฟังก์ชันจะถูกแสดงโดยคอมพิวเตอร์กลาง	ฟังก์ชันจะอินทิเกรตในรูปแบบของกลุ่มผลิตภัณฑ์
5. ความสำคัญของความผิดพลาดจุดเดียว	ต่ำขึ้นกับรูปแบบโครงสร้าง	สูง	ต่ำขึ้นกับรูปแบบโครงสร้าง
6. ราคาในการติดตั้ง	สูงขึ้นกับการเชื่อมต่อสายแยกและเครื่องมือที่มีปริมาณมาก	ปานกลางเนื่องจากสามารถประหยัดในส่วนของห้องควบคุมและที่ว่างสำหรับห้องเครื่องมือแต่ก็ใช้การเชื่อมต่อสายแยกได้	ต่ำเนื่องจากสามารถประหยัดค่าใช้จ่ายในส่วนของเชื่อมต่อสายและที่ว่างสำหรับเครื่องมือ
7. การบำรุงรักษา	ต่ำเนื่องจากมีหลายรูปแบบมากแต่มีอะไหล่ น้อยแบบ	ปานกลางเนื่องจากต้องการบุคคลที่ได้รับการฝึกอบรมอย่างดี ทางด้านการบำรุงรักษาคอมพิวเตอร์	ดีเนื่องจากสามารถเปลี่ยนแปลงแทนที่อะไหล่และรูปแบบโครงสร้างได้อย่างอัตโนมัติ

ตารางที่ 1.1 แสดงการเปรียบเทียบระบบควบคุมแบบต่างๆ

บทที่ 2 ทัศนวิสัยและหลักการ

ทฤษฎี

ประวัติศาสตร์ของความพยายามของมนุษย์ในการควบคุม จะเห็นได้ว่าขบวนการทางอุตสาหกรรมให้เป็นแบบอัตโนมัติได้มีมานานแล้ว ตั้งแต่การสร้างเทอร์โมสแตสสำหรับควบคุมเตาเผาในปี ค.ศ. 1620 แต่ระบบควบคุมเริ่มเป็นรูปเป็นร่างอย่างจริงจังเมื่อประมาณ 50 ปีที่ผ่านมาเนื่อง การที่ระบบควบคุมเริ่มมีการพัฒนาอย่างจริงจังตั้งแต่ปี ค.ศ. 1930 มาจนถึงปัจจุบันนั้น มีตัวกระตุ้นอยู่ 2 ประการ คือ ความต้องการของผู้บริโภค และความก้าวหน้าทางเทคโนโลยีที่ทำให้ระบบควบคุม มีประสิทธิภาพมากยิ่งขึ้น วิวัฒนาการของการพัฒนาของระบบควบคุมทางอุตสาหกรรมนั้น สามารถดูได้จากรูปประกอบที่ 2.1 ซึ่งจากภาพจะเห็นได้ว่าระบบควบคุมที่อยู่ในยุคปัจจุบันนี้คือระบบการควบคุมโดยใช้เทคนิคทางดิจิทัลแบบกระจาย (Distributed Control System) ซึ่งก็คือระบบควบคุมที่เป็นโครงการของกลุ่มข้าพเจ้านั่นเอง



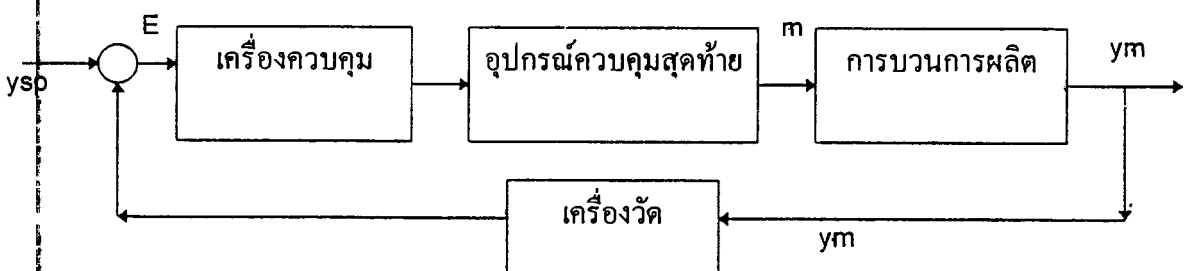
รูปที่ 2.1 แสดงวิวัฒนาการของระบบควบคุม

DCS ย่อมาจากคำว่า " Distributed Control System " เป็นวิวัฒนาการอีกขั้นหนึ่งของระบบควบคุมจากเดิมที่ผู้ควบคุมจะต้องอยู่ใกล้กับตัวคอนโทรลเลอร์ (Controller) ซึ่งมักจะวางใกล้กับกระบวนการที่ถูกทำการควบคุม โดยผู้ควบคุมทำหน้าที่คอยเช็คผล บันทึกข้อมูล และทำการควบคุมจากจุดที่กระบวนการดำเนินอยู่ ซึ่งไม่เป็นการไม่สะดวกเพราะไม่สามารถมองงานระบบรวมและทำการคอนโทรลจากจุดเดียวได้ อีกทั้งในกระบวนการบางอย่างอาจเป็นอันตรายต่อสุขภาพของผู้ที่อยู่ใกล้ได้ ระบบ DCS จึงถือกำเนิดขึ้นมาเพื่อแก้ปัญหาดังกล่าว ด้วยการแบ่งแยกระบบออกให้ส่วนปฏิบัติการ (Operate) และส่วนควบคุมกระบวนการแยกจากกัน ซึ่งระบบ DCS นี้ ประกอบไปด้วยคอมพิวเตอร์หลายๆ ตัวต่อกันเป็นระบบ ฟังก์ชันการควบคุมจะถูกกระจายให้คอมพิวเตอร์เล็กๆ แต่ละส่วน ซึ่งทำหน้าที่ควบคุมกระบวนการย่อยๆ โดยข้อมูลของกระบวนการย่อยๆ ทุกกระบวนการจะถูกส่งผ่านส่วนสื่อสารไปเก็บรวบรวมที่คอมพิวเตอร์อีกตัว ซึ่งทำหน้าที่ติดต่อกับพนักงาน โดยผ่านทางจอภาพและเป็นกด ระบบนี้นิยมใช้เป็นระบบควบคุมอัตโนมัติ สำหรับควบคุมกระบวนการผลิตภายในโรงงานอุตสาหกรรม เช่น เตาเผาและหม้อกำเนิดไอน้ำของโรงจักรไฟฟ้า หอกกลั่น (Distillation Column) ของโรงกลั่นน้ำมัน , ถังปฏิกรณ์ (Reactor Tank) ของโรงงานอุตสาหกรรมเคมี และระบบบำบัดน้ำเสียของโรงงานอุตสาหกรรมต่างๆ

ระบบควบคุมอัตโนมัติในโรงงานอุตสาหกรรมแบบเดิม แบ่งออกเป็น 2 ส่วน คือ การควบคุมแบบป้อนกลับ (Feedback Control) และการควบคุมแบบติดดับ (On-off Control)

การควบคุมแบบป้อนกลับ หรือการควบคุมชนิดสัญญาณต่อเนื่อง (Continuous Control) ภายในโรงงานอุตสาหกรรมแบบเดิม มักติดตั้งเครื่องควบคุมชนิดนิวเมติกส์ หรืออิเล็กทรอนิกส์แบบพีไอดีเป็นเครื่องมือควบคุม

ปัจจุบัน DCS เป็นระบบควบคุมอัตโนมัติ สำหรับกระบวนการผลิตของโรงงานอุตสาหกรรมทั้งหมด DCS สามารถควบคุมกระบวนการผลิตแบบป้อนกลับและกระบวนการผลิตแบบติดดับภายในระบบควบคุมเดียว DCS จึงเป็นระบบควบคุมที่เหมาะสมสำหรับโรงงานอุตสาหกรรมปัจจุบันที่ประกอบด้วยกระบวนการผลิตขนาดใหญ่และซับซ้อน หรือเป็นกระบวนการผลิตแบบเบตซ์ (Batch Process) ที่ต้องการการควบคุมกระบวนการผลิตแบบป้อนกลับ และกระบวนการผลิตแบบติดดับ ภายในกระบวนการผลิตเดียว



รูปที่ 2.2 ระบบควบคุมแบบป้อนกลับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ภายในเท่านั้น ไม่สามารถนำออกไปใช้ประโยชน์ด้านการค้า ไม่สามารถใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การควบคุมแบบป้อนกลับ เป็นการควบคุมพื้นฐานที่เข้าใจง่ายที่มีการส่งสัญญาณป้อนกลับให้กระบวนการ ซึ่งเป็นตัวแปรที่ควบคุมได้ การควบคุมแบบป้อนกลับนี้ สามารถรักษาเสถียรภาพของกระบวนการผลิตเองโดยเครื่องวัด ตรวจสอบ สัญญาณวัดของตัวแปรกระบวนการแสดงสภาพของกระบวนการผลิตเปรียบเทียบกับเป้าหมายการควบคุม โดยเครื่องควบคุมทำหน้าที่ส่งสัญญาณควบคุม (Control Signal) ให้กระบวนการผลิต เพื่อปรับสภาพและรักษาเสถียรภาพของกระบวนการผลิตให้ตรงกับเป้าหมายการควบคุม (Set Point) แต่เนื่องจากสัญญาณของเครื่องควบคุมมักเป็นสัญญาณไฟฟ้า หรือสัญญาณความดันลม ไม่สามารถใช้เปลี่ยนแปลงหรือปรับสภาพของกระบวนการผลิตโดยตรงได้ ดังนั้นสัญญาณของเครื่องควบคุม จึงต้องผ่านอุปกรณ์ควบคุมสุดท้าย (Final Control Elements) เช่น มอเตอร์ไฟฟ้า และวาล์ว เพื่อเปลี่ยนสัญญาณควบคุมเป็นสัญญาณควบคุมกระบวนการที่สามารถปรับสภาพกระบวนการผลิตโดยตรง

จากรูปที่ 2.2 ซึ่งเป็นแผนภาพแสดงระบบการควบคุมแบบป้อนกลับ โดยมีขั้นตอนในการควบคุมสภาพกระบวนการหรือตัวแปรออกของกระบวนการให้มีค่าตามเป้าหมายที่ต้องการ ดังนี้

1. เครื่องวัดจะตรวจสอบสภาพและวัดค่าตัวแปรกระบวนการ Y เช่น อุณหภูมิ ความดัน ได้ค่า y_m ของกระบวนการจากเครื่องวัดที่เป็นตัวแทนแสดงสภาพของกระบวนการ
2. อุปกรณ์เปรียบเทียบ จะมาเปรียบเทียบค่าวัดของกระบวนการ, y_m , กับค่าสภาพกระบวนการที่ต้องการ ซึ่งเป็นเป้าหมายของการควบคุม, y_{sp} , และคำนวณหาความคลาดเคลื่อนในการควบคุม, $E = y_{sp} - y_m$
3. เครื่องควบุนำค่าความคลาดเคลื่อนในการควบคุมมาใช้ในการคำนวณ เพื่อปรับสภาพของกระบวนการให้เข้าสู่เป้าหมายที่ต้องการ โดยส่งสัญญาณควบคุม, C , ไปยังกระบวนการ เพื่อลดค่าความคลาดเคลื่อนในการควบคุม แต่สัญญาณจากเครื่องควบคุมมักจะไม่สามารถใช้ปรับสภาพของกระบวนการได้โดยตรง จึงต้องส่งสัญญาณควบคุมไปยังอุปกรณ์ควบคุมสุดท้าย เพื่อเปลี่ยนรูปหรือเพิ่มขนาดของสัญญาณควบคุมให้มีกำลังมากพอ โดยสัญญาณจากอุปกรณ์ควบคุมสุดท้ายนี้ เรียกว่า สัญญาณปรับกระบวนการ, m

DCS สามารถปฏิบัติหน้าที่การตรวจสอบและควบคุมกระบวนการผลิตภายในโรงงานอุตสาหกรรมโดยหน่วยเครื่องมือภายในหน่วยเชื่อมต่อ และควบคุมกระบวนการของดีซีเอสเป็นผู้ปฏิบัติหน้าที่ตรวจสอบ และควบคุมกระบวนการผลิต เช่นเดียวกับเครื่องมือวัด และเครื่องมือควบคุมของกระบวนการแบบเดิม การกำหนดโครงสร้างของระบบควบคุม DCS คือ การสร้างหน่วยเครื่องมือภายในและกำหนดรายละเอียดการปฏิบัติงาน ตรวจสอบ และควบคุมกระบวนการผลิตของหน่วยเครื่องมือภายในของ DCS คล้ายกับการติดตั้งเครื่องมือวัด และเครื่องมือควบคุมภายในโรงงานอุตสาหกรรมของระบบควบคุมแบบเดิม โดยการเชื่อมโยงสายสัญญาณวัดและสายสัญญาณควบคุมระหว่างเครื่องมือภายในระบบควบคุมของโรงงานอุตสาหกรรม ดีซีเอสอาจกำหนดโครงสร้างของระบบควบคุมโดยการเขียน โปรแกรมเช่นเดียวกับเครื่องคอมพิวเตอร์ทั่วไป แต่การกำหนดโครงสร้างนี้เป็นเอกสารที่สแกนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สร้างของระบบควบคุมโดยการเขียนโปรแกรมคอมพิวเตอร์ไม่เหมาะกับ DCS เนื่องจากวิศวกรผู้เกี่ยวข้องกับการปฏิบัติหน้าที่กำหนดโครงสร้างของระบบควบคุมโดยตรง มักเป็นผู้ขาดความรู้ และไม่มีประสบการณ์เพียงพอ สำหรับการเขียนโปรแกรมคอมพิวเตอร์ เพื่อกำหนดโครงสร้างของระบบควบคุมให้กับ DCS บริษัทผู้ผลิต DCS จึงเลือกวิธีกำหนดโครงสร้างของระบบควบคุมโดยการเชื่อมโยงข้อมูลระหว่างหน่วยเครื่องมือภายในของ DCS เช่นเดียวกับการเชื่อมโยงสายสัญญาณวัด และสายสัญญาณควบคุมเครื่องมือภายในในระบบควบคุมของโรงงาน อุตสาหกรรมแบบเดิม ดังนั้น DCS จึงมีรูปแบบแตกต่างกันไปอยู่หลายลักษณะ แล้วแต่บริษัทที่ผลิต สำหรับโครงการชิ้นนี้ใช้ลักษณะภาพรวมที่เป็นพื้นฐานทั่วไป โดยสามารถแบ่งระบบออกได้เป็น 3 ส่วน ดังนี้

1. Human Interface
2. Communication Interface
3. Controlling Unit

Human Interface ทำหน้าที่ดังนี้

- ติดต่อกับผู้ใช้ให้ป้อนค่าฟังก์ชัน และข้อมูล พารามิเตอร์ต่าง ๆ ของการควบคุม
- แปลงข้อมูลที่ผู้ใช้ป้อนเข้ามาในอยู่ในรูปรหัส เพื่อส่งให้หน่วยคอนโทรลเลอร์ (Controller Unit)
- รับค่าผลลัพธ์จากการปฏิบัติการของ Controlling Unit มาบันทึกและจัดกาให้อยู่ในรูปที่แสดงผลที่ผู้ใช้สามารถเข้าใจได้ง่าย เพื่อนำไปวิเคราะห์หรือปรับปรุงต่อไป

ในส่วน Human Interface นี้ สามารถแบ่งออกเป็น 2 ส่วน ตามลำดับหน้าที่ได้ดังนี้

1. Operator Interface มีไว้สำหรับผู้ควบคุมตรวจสอบสถานะ และควบคุมกระบวนการผลิต
2. Engineering Interface มีไว้สำหรับวิศวกรระบบ ทำหน้าที่ปรับเปลี่ยนหรือเพิ่มเติมระบบควบคุมรวมทั้งระบบ Back up ในกรณีที่เกิดการ Shut down ด้วย

Communication System ทำหน้าที่ในการควบคุมการติดต่อสื่อสารข้อมูลระหว่าง Human Interface กับ Controlling Unit

Controlling Unit มีไว้สำหรับการควบคุมกระบวนการผลิตตามขั้นตอนที่โปรแกรมไว้

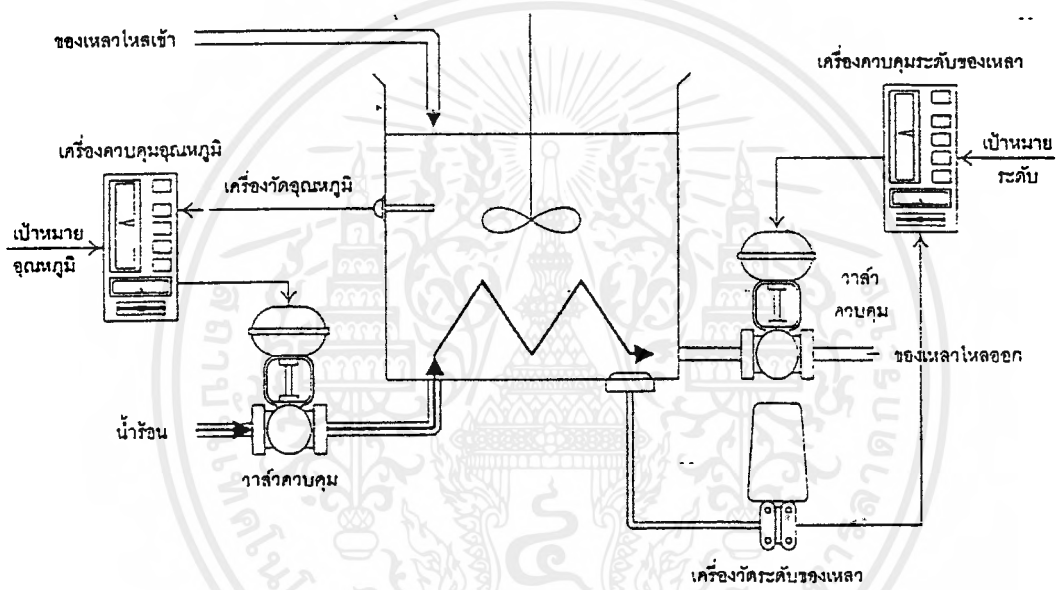
โดยผ่านทาง Engineering Interface ซึ่งตัวแปรบางตัวสามารถเปลี่ยนแปลงได้โดย Operator -

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Interface การควบคุมกระบวนการผลิตของผู้ควบคุม จะต้องอาศัยจอภาพและเป็นกด ซึ่งเราสามารถกำหนดหน้าที่ของ Controlling Unit ได้อย่างคร่าว ๆ ดังนี้

- รับค่าปฏิบัติกรจาก Human Interface
- ประมวลผลและควบคุมกระบวนการจนถึงจุดเป้าหมายที่ตั้งไว้ (Set Point)
- ส่งผลลัพธ์กลับไปให้กับ Human Interface อีกครั้งหนึ่ง ดังจะเห็นได้จาก

ตัวอย่างการควบคุมของระบบถังผสมถ่ายเทความร้อน ดังรูปที่ 2.3



รูปที่ 2.3 ถังผสมถ่ายเทความร้อน

ตัวอย่าง ระบบควบคุมแบบป้อนกลับของถังผสมถ่ายเทความร้อน ประกอบด้วย การควบคุมอุณหภูมิและระดับของเหลว การควบคุมระดับของเหลวต้องการรักษาระดับของเหลวภายในถังผสมถ่ายเทความร้อนให้คงที่ตลอดเวลาจากผลกระทบของสภาพแวดล้อม คือ การไหลเข้าของของเหลวที่เปลี่ยนแปลงตลอดเวลา โดยเครื่องควบคุมระดับของเหลวทำหน้าที่ปรับวาล์วควบคุมให้อัตราการไหลออกของของเหลวสมดุลกับอัตราการไหลเข้าของของเหลวและระดับของเหลวภายในถังผสมถ่ายเทความร้อนให้คงที่ตลอดเวลา จากผลกระทบของอุณหภูมิและอัตราการไหลเข้าของของเหลว โดยเครื่องควบคุมอุณหภูมิทำหน้าที่ปรับวาล์วควบคุมน้ำร้อนให้ความร้อนเข้าสู่ถังผสมถ่ายเทความร้อนสมดุลกับอุณหภูมิและอัตราการไหลเข้าของของเหลว และอุณหภูมิของของเหลวภายในถังผสมถ่ายเทความร้อนตลอดเวลา

หลักการ

รูปแบบการส่งข้อมูลในระบบ

1. การส่งแบบ Broadcast มีลักษณะดังต่อไปนี้ คือ เป็นการส่งข้อมูลที่ไม่มีการกำหนด Address เป้าหมายที่แน่นอน ข้อมูลจะถูกส่งกระจายไปยังหลายๆ อุปกรณ์ในเวลาเดียวกัน โดยมีรอบการส่งที่แน่นอน

ข้อมูลที่ถูกส่งกระจายไปในลักษณะนี้มี 3 ชนิดดังนี้

- Time Interval Cycle (TIC) มีไว้เพื่อเป็นตัวจัดเตรียมสัญญาณเวลาที่สอดคล้องกับสัญญาณนาฬิกาของอุปกรณ์ในระบบทั้งหมด

- Link Message เป็นรูปแบบการส่งข้อมูลที่อุปกรณ์ที่จะทำการส่งข้อมูล ไม่จำเป็นต้องรู้ว่าจะต้องส่งข้อมูลไปที่ไหนแต่อุปกรณ์ที่ทำการรับข้อมูลจะทำการตรวจสอบเองว่าข้อมูลใดที่ต้องการ โดยรอบการทำงานของ Link Message จะกระทำจนครบทุกอุปกรณ์ในระบบ

- Link Request Message เป็นรูปแบบการร้องขอให้อุปกรณ์การส่งข้อมูลทำการส่งข้อมูลซ้ำอีกครั้งหนึ่งในกรณีที่การรับข้อมูลเกิดความผิดพลาด

2. การส่งข้อมูลแบบ Point to Point มีลักษณะดังต่อไปนี้ คือ เป็นการส่งข้อมูลที่มีการกำหนด Address เป้าหมายที่แน่นอน ข้อมูลจะถูกส่งไปยังอุปกรณ์ที่เจาะจงใช้กับระบบการติดต่อสื่อสารที่เกิดขึ้นเป็นครั้งคราว เช่น การเกิด Alarm, การเปลี่ยนแปลงค่าต่างๆ โดย Operator และการร้องขอข้อมูลที่ต้องการให้รายงาน

ข้อมูลที่ถูกส่งในลักษณะนี้มีข้อแตกต่างจากการส่งข้อมูลแบบ Broadcast ดังนี้

- อุปกรณ์ที่ทำการรับข้อมูลจะถูกเจาะจงไว้เพียงแต่ตัวเดียว
- ต้องการข้อมูลการรับรู้และข้อมูลตอบกลับ
- ไม่จำเป็นต้องสอดคล้องกับรอบการส่ง (Transmission Cycle)

บทที่ 3 ส่วนประกอบของงานที่ทำและการเปรียบเทียบ

เนื่องจากโครงการชิ้นนี้เป็นโครงการต่อเนื่องทั้งปี ดังนั้นทางคณะผู้จัดทำขออนุญาตินำเสนอส่วนของโครงการที่ทำไป เพื่อที่จะได้เป็นการแสดงให้เห็นได้ชัดเจนถึงลักษณะของงานที่ทำ และขั้นตอนของการทำงาน ซึ่งสามารถแบ่งออกเป็น 3 ส่วนหลัก ๆ ได้ดังนี้

3.1 ส่วนของการเขียนโปรแกรมในการสนับสนุนการติดต่อสื่อสารภายในระบบ

ลักษณะของงานที่ทำเป็นการเขียนโปรแกรมด้วยภาษา C เพื่อใช้เป็นส่วนในการสนับสนุนการติดต่อสื่อสารในระบบควบคุม โดยนำความรู้ในเรื่อง Network Protocol ที่มีชื่อว่า IPX มาประยุกต์ใช้กับรูปแบบการจัดลำดับความสามารถในการส่งข้อมูลแบบ Token Bus

ในขั้นแรก จะขอกล่าวถึง Protocol ชื่อ IPX ก่อนว่า เป็นโปรแกรมการติดต่ออย่างง่ายสำหรับส่งแพคเกจ จากเครื่องควบคุมหนึ่งไปยังเครื่องควบคุมอื่นๆ ซึ่งมีพื้นฐานความเข้าใจในการใช้งานอย่างไร ดังจะสามารถอธิบายได้ดังนี้

พื้นฐานการใช้งาน มีข้อควรปฏิบัติดังนี้

1. ก่อนที่จะสามารถทำอะไรกับ IPX ได้ จะต้องทำการเรียก IPX Initialize () เพราะนี่จะเป็นการบ่งบอกถึงไดรฟ์เวอร์ (Driver) ที่ต้องการจะใช้
2. ต่อจากนั้นจึงทำการเลือกซ็อกเก็ต (Socket) นั้น ถ้าต้องการเพียงแต่ส่งข้อมูลเท่านั้น เราสามารถที่จะให้หมายเลขซ็อกเก็ตเป็น 0 แล้ว IPX จะเป็นตัวค้นหาซ็อกเก็ต และกำหนดหมายเลขให้
3. ในการเลือกซ็อกเก็ตควรเลือกให้สูงกว่า 0 x 8000
4. ในการเปิดใช้ซ็อกเก็ต ให้ทำการเรียก IPX Open Socket () รหัสที่ถูกส่งกลับมาจะเป็นซ็อกเก็ตที่ถูกเปิดใช้ หรือเป็นรหัส ul ในกรณีที่การเปิดใช้ซ็อกเก็ตนั้น เกิดการผิดพลาดขึ้น

การส่งข้อมูล มีข้อควรปฏิบัติดังนี้

1. ในการที่จะส่งข้อมูล จะต้องทำการจัดสรรบัพเฟอร์ที่ใหญ่พอสำหรับ ECB (Event Control Block) และนอกจากนั้นยังต้องจัดบัพเฟอร์ให้กับ IPX header ด้วย
2. ต้องทำการตั้งค่าเริ่มต้นให้กับสมาชิกใน ECB และใน IPX header ก่อนที่จะทำการส่งซ็อกเก็ต

Table 1

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ในการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

void          far *linkAddress;

void          (far *ESRAddress) ();    ***

BYTE         InUseFlag;

BYTE         CompletionCode;

WORD SocketNumber;                    ***//Big - Endian

BYTE         IPXWorkspace[4];

BYTE         DriverWorkspace[12];

WORD FragmentCount;                   ***

ECBFragment  FragmentDescriptor[4];   ***

}ECB;

```

3. จากตารางที่ 1 สามารถบ่งบอกถึงฟิลด์ต่างๆ ที่ต้องการ สำหรับ ECB และเป็นเรื่องที่สำคัญมากที่ Fragment Descriptor [1] จะต้องทำการส่งค่า Pointer ที่ชี้อยู่ที่ข้อมูลที่ต้องการส่งพร้อมด้วยขนาดของข้อมูลที่ต้องการส่งด้วย

Table 2

```

typedef struct IPXHeader {
    WORD      CheckSum;
    WORD      Length;
    BYTE      TransportControl;
    BYTE      PacketType;          ***
    IPXAddress Destination;        ***
    IPXAddress Source;
} IPXHeader;

```

4. จากตารางที่ 2 แสดงฟิลด์ต่างๆ ที่ต้องการของ IPX header และต้องคำนึงว่าค่าสำหรับใน Packet Type ควรจะเก็บเป็น 4 เสมอ

5. ถ้าต้องการส่งข้อมูลกระจายไปทั่วถึงทุกๆ ที่ ในเครือข่ายต้องใส่ค่าลงใน Immediate Address และที่อยู่ของเป้าหมายด้วยค่า 0XFF ในทุก ๆ ไบต์ของ fields ที่อยู่ในตารางที่ 2

6. เพื่อที่จะได้มาซึ่งค่าที่จะใส่ลงใน Immediate Address ตามข้อ 5 จะต้องทำการเรียก IPX GetLocalTarget() พร้อมด้วยที่อยู่ของ IPX และใช้ Pointer ชี้ไปที่ Immediate Address

หลังจากนั้นมันจะจัดการใส่ค่าให้การใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7. เมื่อ ECB และ Packet header สมบูรณ์แล้ว เราสามารถจะเรียกใช้ IPXSendPacket() พร้อมกับ Pointer เพื่อเติมใส่ลงใน ECB สามารถทำให้โปรแกรมสามารถที่จะดู CompletionCode ของ ECB ได้ทันที และใช้ตัวแปร InUseFlag ในการที่จะรู้ว่าเมื่อใดที่แพคเกจได้ถูกทำการส่ง ซึ่งขึ้นกับการนำไปประยุกต์ใช้งานว่าต้องการจะทำให้หน่วยความจำในส่วนนี้ว่างหรือไม่ หรือจะใช้มันอีกในการส่งครั้งต่อไป

8. หลังจากได้ทำการทดลองในข้อ 1 - 7 แล้ว ไดรฟ์เวอร์ (Driver) ของ IPX จะทำการส่งข้อมูลได้ง่ายดาย แล้วจะไปทำการเปลี่ยนค่าของรหัส Completion ในตารางที่ 3 และ 4 เพื่อทำการจัดค่าให้กับ InuseFlag และ CompletionCode ต่อไป

Table 3

InUse Flag Values

00h	Available
F8h	IPX in critical section
FAh	Processing
FBh	Holding
FEh	Awaiting packet reception
FFh	Sending packet

Table 4

Completion Code Values

00h	Success
ECh	Remote terminated connection without acknowledging packet
EDh	Abnormal connection termination
EEh	Invalid connection ID
F9h	Event should not be cancelled
FAh	Cannot establish connection with specified destination
FCh	Cancelled
FDh	Malformed packet
FEh	Packet undeliverable
FFh	Physical error

การรับข้อมูล

1. ในการที่จะรับข้อมูล จะต้องทำการจัดสรรหน่วยความจำให้กับ ECB , IPX header และข้อมูลใดๆ ที่ต้องการจะรับและต้องกำหนดให้ ESR ด้วยแต่ปกติจะทำการเซตค่า ESR เป็น 0 (NULL)
2. หลังจากนั้นทำการเรียกฟังก์ชัน IPX ListenForPacket() พร้อมกับ Pointer ไปที่ ECB ซึ่งเป็นที่ที่ไดร์ฟเวอร์ของ IPX จะใส่ค่าข้อมูลที่รับลงไป ใน IPX header
3. ตอนนี้สามารถดูค่า CompletionCode ของ ECB ที่เตรียมไว้ว่า เมื่อไรที่มีค่า 0X00 สามารถที่จะดึงข้อมูลกลับมา แล้วนำ ECB ไปใช้ใหม่ สำหรับข้อมูล หรือแพคเกจที่จะเข้ามาต่อไป
4. กรณีที่กำลังรับข้อมูลมาก ๆ ควรจะจัดเตรียมให้มี ECB มากกว่าหนึ่ง เนื่องจากถ้า ECB ไม่เพียงพอก็จะไม่สามารถรับแพคเกจที่ส่งมาได้หมด นั่นหมายความว่าความไม่สมบูรณ์ของข้อมูลด้วย

นอกเหนือจากความรู้เกี่ยวกับ IPX ดังที่ได้กล่าวมาในข้างต้นแล้ว ยังต้องการความรู้ทางด้านการจัดลำดับความสามารถในการส่งข้อมูลใน LAN อีกด้วย ซึ่งในโครงการนี้ได้เลือกใช้การจัดลำดับแบบ Token Bus ดังรายละเอียดต่อไปนี้

Token Bus เป็นรูปแบบของการจัดลำดับการส่งข้อมูลอย่างหนึ่ง ซึ่งในโครงการนี้ได้นำตารางแสดงการเปรียบเทียบข้อดีและข้อเสียของระบบ Token Bus กับระบบของการจัดลำดับข้อมูลอย่างอื่น ดังแสดงดังตารางที่ 3.1

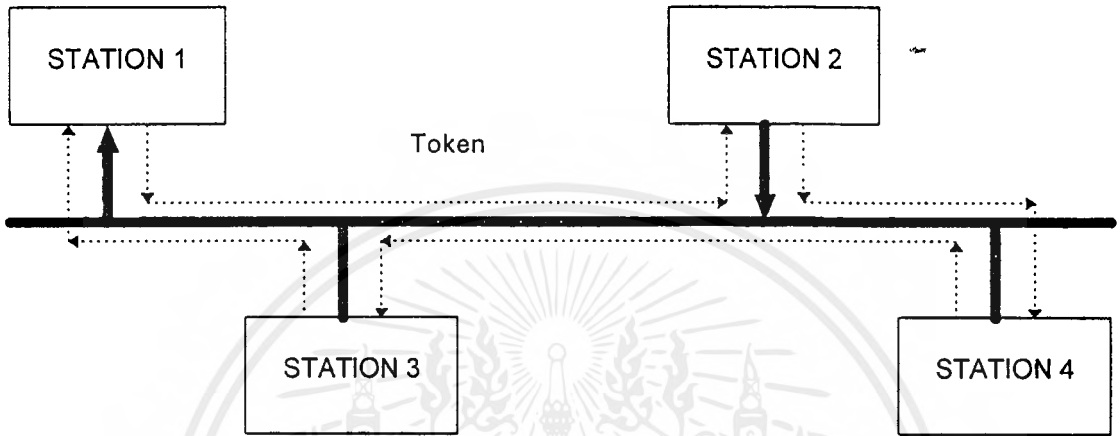
จากตารางจะเห็นว่าระบบ Token Bus มีข้อดี คือ ในการส่งผ่านข้อมูลจะต้องมีการรอรับสิทธิ์ในการส่งและรับข้อมูล (Token) จึงสามารถทำให้การส่งและรับข้อมูลเป็นไปอย่างมีระบบ

รูปแบบ	ลักษณะการใช้งาน	ความสามารถในการส่งข้อมูล	ข้อดี	ข้อเสีย
CSMA/CD	ส่วนใหญ่ใช้ร่วมกับ การสื่อสารแบบ Ethernet	อัตราในการส่งข้อมูลจะลดลงเมื่อมีการเพิ่มขึ้นของ Traffic ในระบบ	ลดเวลาที่เกิดขึ้นจากการชนกันเมื่อเปรียบเทียบกับระบบที่ไม่มี Collision Detect	ไม่สามารถระบุเวลาที่แน่นอนในการส่งข้อมูล 1 Frame ให้สำเร็จ
TOKEN RING	นิยมใช้ในงานบริษัทและงานทางธุรกิจ	อัตราในการส่งข้อมูลจะมีความแน่นอน	มีการ Interface ที่ง่ายในการส่งผ่านข้อมูลโดยต้องมีการรอ Token	ถ้ามีเครื่องใดเครื่องหนึ่งเสียจะทำให้ระบบทั้งระบบเกิดความเสียหาย
TOKEN BUS	นิยมใช้ในการควบคุมแบบอัตโนมัติ	อัตราในการส่งข้อมูลจะมีความแน่นอน	ในการส่งผ่านข้อมูลโดยต้องมีการรอรับ Token	การเพิ่มหรือลดจำนวนเครื่องในระบบทำได้ยากเพราะจะต้องบอกเครื่องข้างเคียงให้รับรู้ถึงการเปลี่ยนแปลงนั้น

ตารางที่ 3.1 แสดงรูปแบบการจัดลำดับความสามารถในการส่งข้อมูลใน LAN



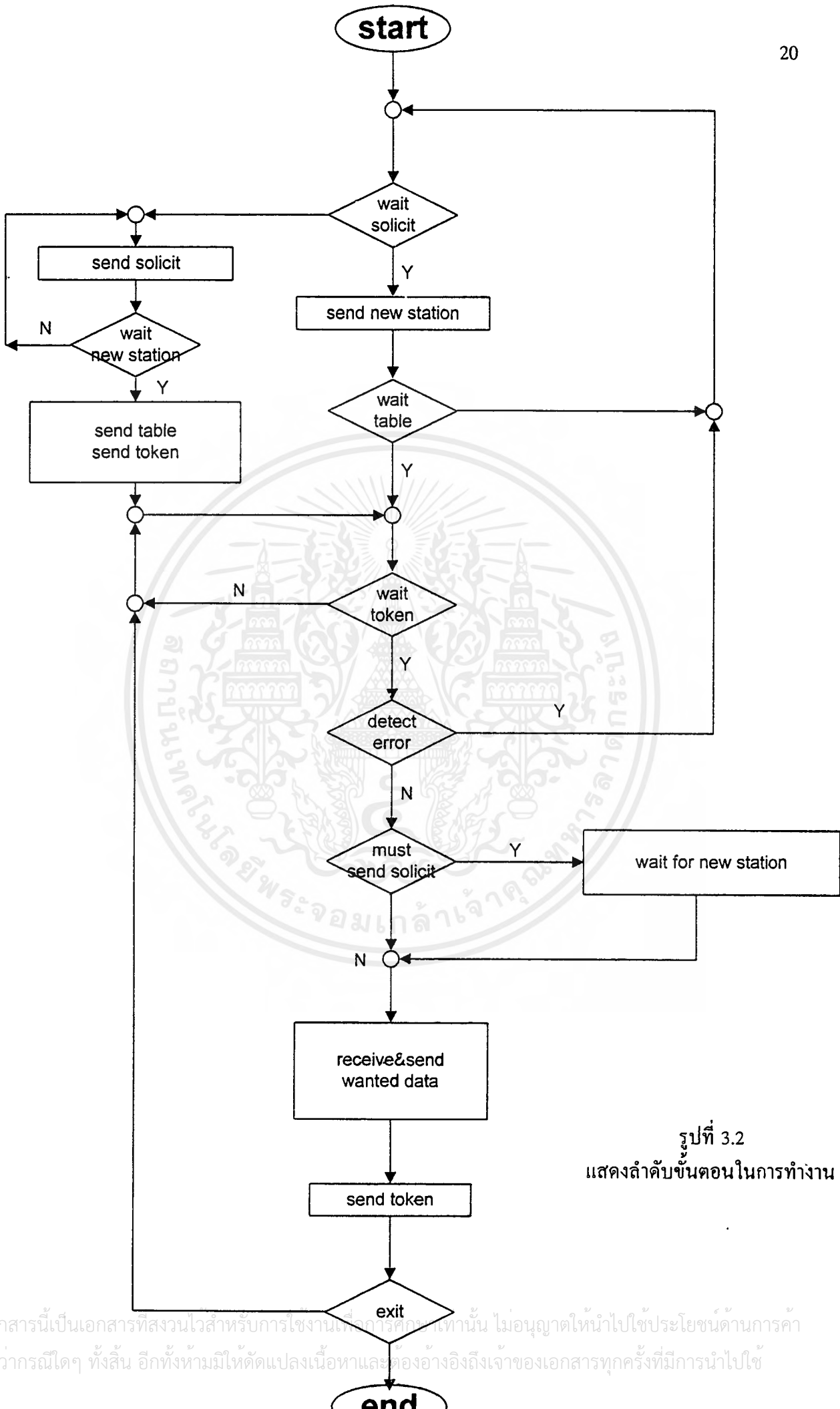
ซึ่งจากการพิจารณาแล้วพบว่า เป็นเรื่องที่ดีที่สุดในการนำมาเป็นส่วนหนึ่งในการทำการสนับสนุนทางด้านการติดต่อสื่อสารในระบบคอมพิวเตอร์ของคณะผู้จัดทำ เนื่องจากสามารถหาอุปกรณ์ในการสนับสนุน Protocol ชนิดนี้ได้ง่ายและสามารถเลียนแบบการติดต่อสื่อสารทั้งแบบ Broadcast และ Point to Point ได้ใกล้เคียงระบบที่ใช้จริงในวงการอุตสาหกรรมมากที่สุด



รูปที่ 3.1 Token Bus

จากรูปที่ 3.1 เป็นแผนภาพแสดงการทำงานของระบบ Token Bus ที่ใช้ในการรับส่งข้อมูลเพื่อประโยชน์ในการติดต่อสื่อสาร

จะเห็นได้ว่าจากความรู้ในเรื่องของ Protocol แบบ IPX และความรู้ทางด้านรูปแบบการทำงานของระบบจัดลำดับข้อมูล แบบ Token Bus ที่ได้นำเสนอมาในข้างต้น ประกอบกับความรู้ทางด้านการเขียนโปรแกรมโดยใช้ภาษา C ทำให้สามารถสร้างซอฟต์แวร์ที่มีส่วนช่วยในการสนับสนุนการติดต่อสื่อสารภายในระบบควบคุมได้ ซึ่งสามารถนำมาเขียนเป็น Flow Chart แสดงลำดับขั้นตอนในการคิดการทำงานได้ดังรูปที่ 3.2



รูปที่ 3.2 แสดงลำดับขั้นตอนในการทำงาน

3.2 ส่วนของการเพิ่มโปรแกรมสนับสนุนการติดต่อสื่อสารเข้าไปในโปรแกรมหลัก

การนำเอาโปรแกรมเพิ่มเติมเข้าไปในโปรแกรมหลัก ซึ่งเป็นโปรแกรมส่วนที่ ได้จัดทำขึ้นมาก่อนหน้านี้โดย (1) เพื่อให้โปรแกรมของระบบควบคุมกลายเป็นโปรแกรมเลียนแบบระบบควบคุมแบบกระจายส่วน (DCS) อย่างสมบูรณ์ในจุดที่สามารถควบคุมเป็นโครงข่ายได้

ลำดับขั้นตอนในการนำไปเพิ่มเติมในโปรแกรมหลักนั้น จะขึ้นอยู่กับลักษณะเงื่อนไขเดิมต่างๆ ของโปรแกรมหลัก จึงต้องทำการแก้ไขเปลี่ยนแปลงตัวแปร และเชื่อมโยงให้โปรแกรมสามารถที่จะทำงานร่วมกันกับโปรแกรมหลัก หลังจากทำการเพิ่มเติมเรียบร้อยแล้ว ในทุก ๆ หน้าจอของโปรแกรมใหม่ จะมีคำว่า " NODE " เพิ่มขึ้นมาจากหน้าจอในกรณีโปรแกรมเดิม ซึ่งหมายความว่า สามารถใช้ระบบควบคุมนี้ ทำการควบคุมได้โดยภายในระบบประกอบด้วยเครื่องควบคุมหลายตัวที่ควบคุมระบบหลายระบบได้ จากของเดิมซึ่งจะต้องใช้ควบคุมระบบเพียงระบบเดียว และด้วยเครื่องควบคุมตัวเดียวเท่านั้น อีกส่วนหนึ่งที่เปลี่ยนแปลงคือในส่วนของการใช้หน้าจอวาดภาพคือสามารถที่จะวาดภาพได้ ขณะที่ทำการ Run Mode

3.3 ส่วนของการสร้างโปรแกรมในส่วนของการสนับสนุนการใช้เมาส์

แล้วนำไปเพิ่มใส่เข้าไปในโปรแกรมตัวเดิม หลังจากเพิ่มเติมโปรแกรมสนับสนุนการใช้เมาส์ ในการเข้าถึงจุดต่างๆ ของหน้าจอได้แทนการใช้คีย์บอร์ดของโปรแกรมเดิม ทำให้เกิดความสะดวกในการใช้โปรแกรมมากขึ้น ต่อจากนี้จะเป็น Source Code ของโปรแกรมสนับสนุน การใช้เมาส์ ซึ่งสามารถแสดงให้เห็นถึงฟังก์ชันต่างๆ ที่นำมาใช้ และขั้นตอนการคิดได้อย่างชัดเจน

```
#include <dos.h>
#include <stdio.h>
#include <stdlib.h>
#include <conio.h>
#include <memory.h>
#include <graph.h>
#include <string.h>
#define Left_button 0;
#define Middle_button 4;
#define Right_button 2;
#define BLACK 0;
#define GREEN 2;
#define WHITE 7;
int MouseRetall;
int mouse_initialize(void);
void mouse_show_cursor(void);
void mouse_hide_cursor(void);
void set_mouse_pixel_ratio(int hor,int ver); // Set movement speed of mouse
void set_mouse_double_threshold(int threshold);
void mouse_button(int *L_button,int *R_button,int *M_button,int *x,int *y);
leftb_pressed(void);
rightb_pressed(void);
middleb_pressed(void);
void mouse_position(int *x,int *y);
void set_mouse_position(int x,int y);
void mouse_horizontal_range(int left,int right);
void mouse_vertical_range(int top,int bottom);
void mouse_times_pressed(int mbutton,int *num_times,int *x,int *y);
void mouse_times_release(int mbutton,int *num_times,int *x,int *y);
void mouse_move_corsor(int row,int col);
void mouse_set_page(page)
```

```

int page:
{
    union REGS regs;
    if (MouseInstall==1) {
        regs.x.ax = 0x001D;
        regs.x.bx = page;
        int86(0x33,&regs,&regs);
    }
}
int mouse_get_page(void)
{
    union REGS regs;
    if (MouseInstall==1) {
        regs.x.ax = 0x001E;
        int86(0x33,&regs,&regs);
        return regs.x.bx;
    } else return -1;
}
int mouse_initialize(void)
{
    union REGS regs;
    regs.x.ax = 0;
    int86(0x33,&regs,&regs);
    return regs.x.ax;
}
void mouse_show_cursor(void)
{
    union REGS regs;
    if (MouseInstall==1) {
        regs.x.ax = 1;
        int86(0x33,&regs,&regs);
    }
}
void mouse_hide_cursor(void)
{
    union REGS regs;
    if (MouseInstall==1) {
        regs.x.ax = 2;
        int86(0x33,&regs,&regs);
    }
}
void set_mouse_pixel_ratio(int hor,int ver) /* Set movement speed of mouse */
{
    union REGS regs;
    regs.x.ax = 0x000F;
    regs.x.cx = hor;
    regs.x.dx = ver;
    int86(0x33,&regs,&regs);
}
void set_mouse_double_threshold(int threshold)
{
    union REGS regs;
    regs.x.ax = 0x0010;
    regs.x.dx = threshold;
    int86(0x33,&regs,&regs);
}
void mouse_condition_off(int x1,int y1,int x2,int y2)
{
    union REGS regs;
    regs.x.ax = 0x0010;
    regs.x.cx = x1;
    regs.x.dx = y1;
    regs.x.si = x2;
    regs.x.di = y2;
    int86(0x33,&regs,&regs);
}
void mouse_button(int *L_button,int *R_button,int *M_button,int *x,int *y)
{
    int temp;
    union REGS regs;
    regs.x.ax = 3;
    int86(0x33,&regs,&regs);
}

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ การศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        *x = regs.x.cx;
        *y = regs.x.dx;
        temp = regs.x.bx;
        *R_button = temp & 2;
        temp=regs.x.bx;
        *L_button = temp & 1;
        temp=regs.x.bx;
        *M_button = temp & 4;
    }
leftb_pressed(void)
{
    union REGS regs;
    regs.x.ax = 3;
    int86(0x33,&regs,&regs);
    return regs.x.bx & 1;
}
rightb_pressed(void)
{
    union REGS regs;
    regs.x.ax = 3;
    int86(0x33,&regs,&regs);
    return regs.x.bx & 2;
}
middleb_pressed(void)
{
    union REGS regs;
    regs.x.ax = 3;
    int86(0x33,&regs,&regs);
    return regs.x.bx & 4;
}
int Button_pressed(int Button) // 0 = left, 1 = right, 2 = middle
{
    union REGS regs;
    regs.x.ax = 5;
    regs.x.bx = Button;
    int86(0x33,&regs,&regs);
    return regs.x.bx;
}
int Button_release(int Button) // 0 = left, 1 = right, 2 = middle
{
    union REGS regs;
    regs.x.ax = 6;
    regs.x.bx = Button;
    int86(0x33,&regs,&regs);
    return regs.x.bx;
}
mouse_set_cursor(sm_char,sm_attr,cm_char,cm_attr)
char cm_char,cm_attr;
{
    union REGS regs;
    regs.h.ch=sm_attr;
    regs.h.dh=cm_attr;
    regs.h.cl=sm_char;
    regs.h.dl=cm_char;
    regs.x.ax=0xa;
    regs.x.bx=0;
    int86(0x33,&regs,&regs);
}
void mouse_position(int *x,int *y)
{
    union REGS regs;
    regs.x.ax = 3;
    int86(0x33,&regs,&regs);
    *x = regs.x.cx;
    *y = regs.x.dx;
}
void set_mouse_position(int x,int y)
{
    union REGS regs;
    regs.x.ax = 3;
    regs.x.cx = x;
    regs.x.dx = y;
    int86(0x33,&regs,&regs);
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

void mouse_horizontal_range(int left,int right)
{
    union REGS regs;
    regs.x.ax = 7;
    regs.x.cx = right;
    regs.x.dx = left;
    int86(0x33,&regs,&regs);
}
void mouse_vertical_range(int top,int bottom)
{
    union REGS regs;
    regs.x.ax = 8;
    regs.x.cx = top;
    regs.x.dx = bottom;
    int86(0x33,&regs,&regs);
}
void mouse_times_pressed(int mbutton,int *num_times,int *x,int *y)
{
    union REGS regs;
    regs.x.bx = mbutton;
    regs.x.ax = 5;
    int86(0x33,&regs,&regs);
    *num_times = regs.x.bx;
    *x = regs.x.cx;
    *y = regs.x.dx;
}
mouse_move_cursor(int row,int col)
{
    union REGS regs;
    regs.x.dx=8*(row-1);
    regs.x.cx=8*(col-1);
    regs.x.ax=4;
    int86(0x33,&regs,&regs);
}
void mouse_times_release(int mbutton,int *num_times,int *x,int *y)
{
    union REGS regs;
    regs.x.bx = mbutton;
    regs.x.ax = 6;
    int86(0x33,&regs,&regs);
    *num_times = regs.x.bx;
    *x = regs.x.cx;
    *y = regs.x.dx;
}

```

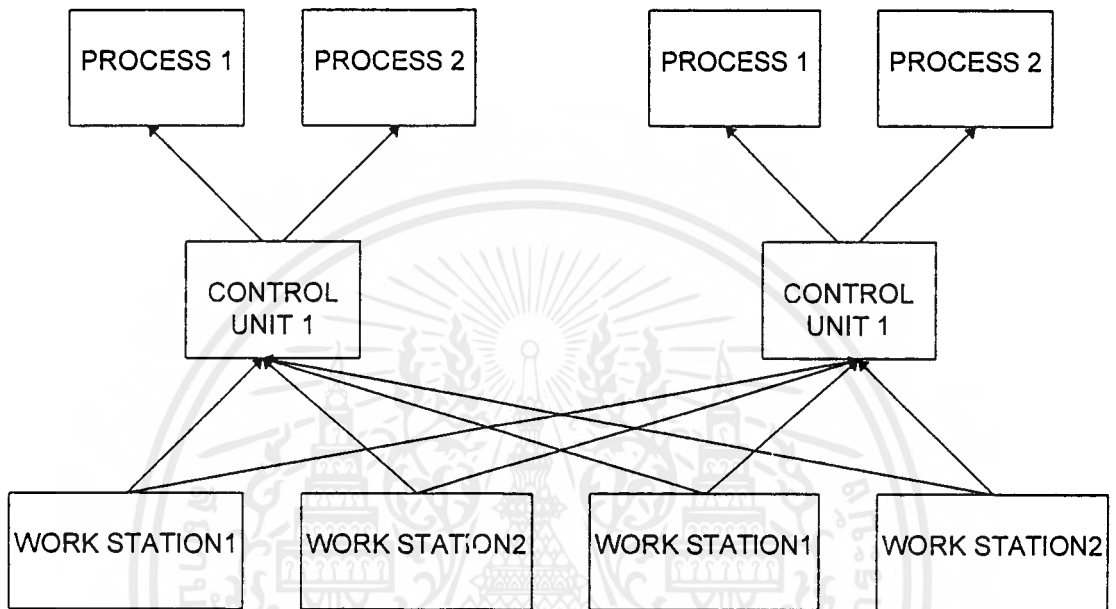
ข้อแตกต่างระหว่างระบบที่เพิ่มเติมส่วนสนับสนุน กับระบบเดิม

1. สามารถใช้เมาส์ในการเข้าถึงโปรแกรม ณ จุดต่างๆได้ แทนการใช้คีย์บอร์ด ซึ่งข้อแตกต่างข้อนี้ ช่วยให้เกิดความสะดวก และง่ายแก่ผู้ใช้
2. สามารถใช้ควบคุมขบวนการหลาย ๆ ขบวนการในเวลาเดียวกันได้ ซึ่งแต่ก่อนระบบเดิมจะสามารถควบคุมขบวนการได้ แต่เพียงขบวนการเดียวต่อเครื่องควบคุมหรือคอมพิวเตอร์หนึ่งเครื่อง หรืออาจจะกล่าวได้ว่าระบบควบคุมตัวใหม่ที่ได้รับการเพิ่มเติมในส่วนที่เป็น โปรแกรมสนับสนุนภาคติดต่อสื่อสารแล้วนั้น สามารถที่จะควบคุมในลักษณะเป็น Network Control ได้นั่นเอง
3. Gedit ต้องทำในขณะที่ระบบทำงานอยู่ ในโปรแกรมระบบเดิม ไม่จำเป็นต้องทำขณะที่Runระบบ

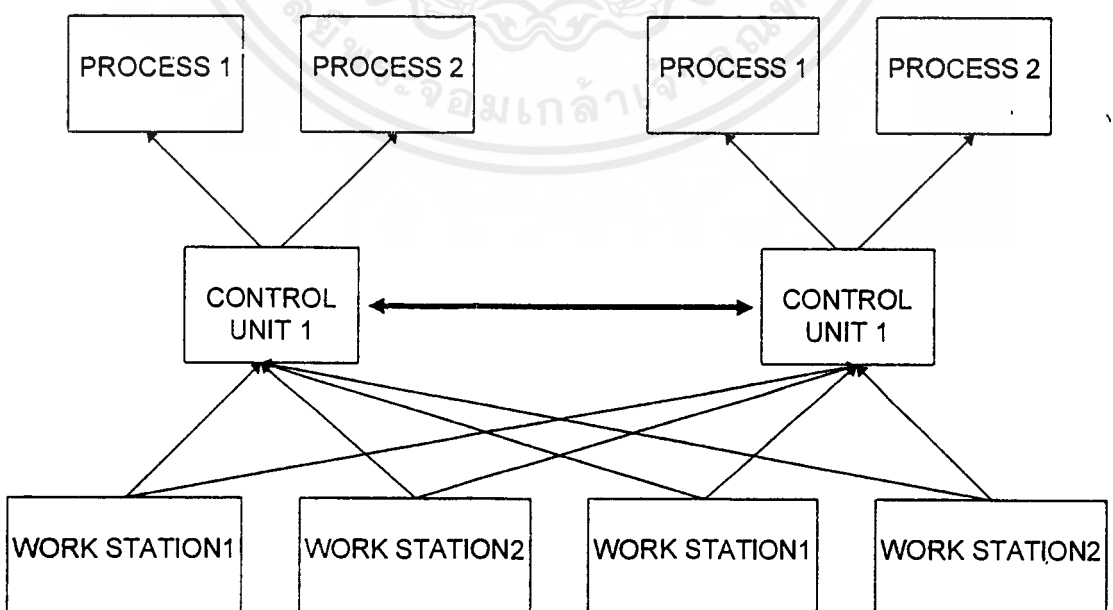
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อแตกต่างระหว่างระบบควบคุม DCS ที่ถูกพัฒนาแล้ว (โรงงานชั้นนำ) กับระบบควบคุม DCS ที่ใช้จริงในวงการอุตสาหกรรม

1. ระบบควบคุม DCS ของคณะผู้จัดทำ ไม่สามารถที่จะควบคุมข้าม Control Unit ได้ แต่ในระบบที่ใช้จริงสามารถทำได้ ดังจะเห็นถึงความแตกต่างได้ ดังรูปที่ 3.3 และ 3.4 ได้



รูปที่ 3.3 แสดงระบบควบคุม DCS ที่เป็นโครงการของคณะผู้จัดทำ



รูปที่ 3.4 แสดงระบบควบคุม DCS ที่ใช้จริงในวงการอุตสาหกรรม

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อใช้ในการศึกษาเท่านั้น ไม่อนุญาตให้เผยแพร่ไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. ระบบที่คณะผู้จัดทำได้พัฒนาขึ้น ยังไม่สามารถจำกัด Priority ของผู้ใช้ในการเข้าถึงส่วนต่างๆ ในแต่ละส่วนของระบบเพื่อเหตุผลของความปลอดภัยได้ แต่ในระบบที่ใช้จริงสามารถทำได้ โดย จะแบ่งระดับความสำคัญของผู้ใช้ หรือชนิดของ Work Station ในการที่จะใช้ควบคุมระบบได้ดังนี้

- View Only เป็น Work station ที่สามารถใช้ได้แต่การดูค่าเพียงอย่างเดียว ไม่สามารถเซตค่าได้
- Process Engineering Functions เป็นการแบ่งระดับการเข้าถึงในระดับวิศวกรระบบ
- Process Operating Functions เป็นการแบ่งระดับการเข้าถึงในระดับผู้ดูแลระบบ
- Maintenance Technician's Function เป็นการแบ่งระดับการเข้าถึงในระดับช่างเทคนิค

3. อุปกรณ์อำนวยความสะดวกต่างของระบบที่คณะผู้จัดทำได้ทำขึ้นยังมีน้อยมาก เมื่อเทียบกับระบบที่ใช้จริงในวงการอุตสาหกรรม เช่น Cartridge Disk Drive , Dot Matrix Printer., Touch Screen และ Trend Pens

4. รูปแบบของอุปกรณ์ต่างๆ ภายในระบบของคณะผู้จัดทำแตกต่างกับระบบที่ใช้จริง ในแง่ของการเชื่อมต่อความสะดวกในการใช้งาน ความมีลักษณะเฉพาะตัวในการใช้งาน และความสวยงาม เช่น รูปแบบของมอนิเตอร์ และคีย์บอร์ดของระบบที่ใช้งานจริงนั้น ได้ถูกออกแบบให้มีลักษณะเฉพาะเพื่อใช้ในการควบคุมระบบเลยโดยตรง แต่ของคณะผู้จัดทำนั้นเป็นการนำเครื่อง PC มาประยุกต์

5. ฟังก์ชันอำนวยความสะดวกต่อการปฏิบัติงานต่างๆ ของระบบที่ใช้งานจริงนั้น มีมากกว่าระบบที่คณะผู้จัดทำได้ทำขึ้น เช่น ฟังก์ชันของ HELP ซึ่งจะช่วยให้ผู้ใช้ปฏิบัติงานได้สำเร็จ ลุล่วงดียิ่งขึ้น , ฟังก์ชันสำหรับใช้ในการพิมพ์ข้อมูลที่ต้องการ

6. ระบบที่คณะผู้จัดทำได้ทำขึ้นนั้น ไม่มีในส่วนของระบบ Redundance ที่จะใช้สำหรับช่วยในการ Back up ระบบ แต่ในระบบที่ใช้งานจริงมีทั้งแบบ ระบบ Redundance ภายใน และ ภายนอก

7. ระบบที่คณะผู้จัดทำได้ทำขึ้นนั้น ไม่ได้เตรียมการสนับสนุน ให้ใช้กับระบบที่มีจำนวน Work Station มากๆ เนื่องจากความเร็วที่ใช้ในการส่งข้อมูลของระบบที่คณะผู้จัดทำได้ทำขึ้นมีความเร็วประมาณ 2 - 3 MBITS ซึ่งระบบที่ใช้จริงในโรงงานอุตสาหกรรมมีความเร็วในการส่งข้อมูลที่สูงกว่านี้มาก

8. การติดต่อสื่อสารในระบบของคณะผู้จัดทำยังขาดในส่วนของ Time Interval Cycle ซึ่งได้กล่าวถึงมาแล้วในส่วนของหลักการที่อยู่ในบทที่ 2

บทที่ 4 ลักษณะโดยรวมของระบบ

Human Interface

1. OPERATOR CONSOLE

- เพื่อที่จะทำให้เกิดความพึงพอใจแก่ผู้ใช้ระบบ จึงจัดให้มีทางเลือกด้าน Hardware Station มากมายหลายทางเลือก ซึ่งในแต่ละกรณีจะมีเครื่อง PC ที่ประกอบด้วย จอภาพสีหรือขาวดำ และคีย์บอร์ด เป็นเครื่องมือเบื้องต้นในการช่วยให้ผู้ใช้สามารถติดต่อกับระบบ และติดต่อกันภายในระบบควบคุมของตัวมันเองได้อีกด้วย ซึ่งถ้าต้องการจะมองภาพในแง่ของห้องควบคุม จะสามารถเห็นได้ว่าจะต้องมี Station หลาย ๆ ตัวในระบบ

2. SOFTWARE

- โปรแกรมได้ถูกพัฒนาขึ้นตามลำดับตามที่ได้อธิบายไปแล้วในบทที่ 3 เรื่องส่วนหลัก ๆ ของการทำงานของคณะผู้จัดทำ โปรแกรมที่ถูกจัดทำขึ้นได้ทำการถูกออกแบบให้ง่ายต่อการใช้งาน เช่น มีการใช้การกดคีย์บอร์ดไม่ซับซ้อนมากนัก และต่อมาโปรแกรมได้ถูกพัฒนาให้สามารถใช้เมาส์ในการเลือกตั้งค่าฟังก์ชันต่าง ๆ หรือเลือกหน้าจอแสดงผลต่างๆ ได้โดยง่ายด้วยการกดปุ่ม Click บนเมาส์เท่านั้น

3. VIDEO DISPLAYS

ชนิดของส่วนแสดงผล

จอแสดงผลในโครงการชิ้นนี้ เป็นจอมอนิเตอร์ของ PC ธรรมดา ซึ่งส่วนแสดงผลได้ถูกออกแบบให้สามารถแสดงข้อมูลให้กับผู้ใช้ได้ในทางที่ดีที่สุดเท่าที่เป็นไปได้ ส่วนแสดงผลหลายส่วนประกอบด้วย 2 ส่วน คือ ส่วนแรกเป็นส่วนที่ข้อมูลถูกทำการ update อยู่ตลอดเวลา หรือส่วนที่รูปภาพสามารถเปลี่ยนได้ตามค่าของระบบ เช่น สีของน้ำสามารถเปลี่ยนสีได้เมื่อถึงจุดเป้าหมายที่ต้องการ อีกส่วนเป็นส่วนที่สามารถรับอินพุตจากผู้ใช้ และส่งผ่านไปให้ส่วนอื่น ๆ ของระบบควบคุม เช่น ในหน้าจอของ Group Display ผู้ใช้สามารถเปลี่ยนค่าของระบบต่าง ๆ รวมทั้งสามารถเปลี่ยนโหมดการควบคุมได้อีกด้วย ซึ่งการเข้าถึงในส่วนนี้ อาจใช้คีย์บอร์ด หรือ เมาส์ดังที่จะได้บรรยายต่อไป

การเรียกส่วนแสดงผล

1. ใช้คีย์บอร์ด โดยการกดปุ่มพื้นฐาน เช่น Tab , shift-Tab , Page up ,Page down , Enter เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. ใช้การกดปุ่มบนเมาส์ ข้อนี้เป็นส่วนที่ทางผู้จัดทำได้ จัดทำเพิ่มเติมเข้าไปเพื่อความสะดวกต่อการเข้าถึงมากกว่าระบบควบคุมเดิม เพียงแค่ลากเมาส์เลื่อนรูปลูกศรไปตรงจุดที่ต้องการจะเข้าถึงแล้วกดปุ่มบนเมาส์ ก็จะทำได้ในส่วนแสดงผลในส่วนที่ต้องการออกมาอย่างง่ายดาย

MODE POSITION	UNIT1 DECIMAL	SYSTEM MEMORY			INPUT DATA
		HEXADECIMAL	BINARY		
I00	0	0000	0000	0000 0000 0000 0000	
I01	0	0000	0000	0000 0000	
I02	0	0000	0000	0000 0000	MOUSE CURSOR
I03	0	0000	0000	0000 0000	
I04	0	0000	0000	0000 0000 0000 0000	
I05	0	0000	0000	0000 0000 0000 0000	
I06	0	0000	0000	0000 0000 0000 0000	
I07	0	0000	0000	0000 0000 0000 0000	
I08	0	0000	0000	0000 0000 0000 0000	
I09	๒	000๒	0000	0000 0000 0000 0000	
I10	0	0000	0000	0000 0000 0000 0000	
I11	0	0000	0000	0000 0000 0000 0000	
I12	0	0000	0000	0000 0000 0000 0000	
I13	0	0000	0000	0000 0000 0000 0000	
I14	0	0000	0000	0000 0000 0000 0000	
I15	0	0000	0000	0000 0000 0000 0000	
I16	0	0000	0030	0000 0000 0000 0000	
I17	0	0000	0000	0000 0000 0000 0000	
I18	0	0000	0000	0000 0000 0000 0000	
I19	0	0000	0000	0000 0000 0000 0000	

รูปที่ 4.1 แสดงหน้าจอตัวอย่างที่มีการใช้เมาส์

4. WORK STATION

ในโครงการนี้ใช้ PC ธรรมดาเป็น Work Station และคีย์บอร์ด ซึ่งในส่วนขอระบบที่ใช้งานจริง จะมีลักษณะและรูปแบบจำเพาะกว่า

หน้าจอ Operating System

1. Initial Screen (หน้าจอเมนูหลัก)

Command :

Computer Control Development System
 Suthian Kiatsunthorn
 Department of Control Engineering
 King Mongkut's Institute of Technology
 Ladkrabang, Bangkok, Thailand 10520
 Tel. 02-3269989

รูปที่ 4.2 แสดงหน้าจอเมนูหลัก

ประโยชน์ ใช้ในการอำนวยความสะดวกในการเข้าถึงหน้าจอต่าง ๆ ให้กับผู้ใช้ ซึ่งจะเรียงลำดับเมนูตามตัวอักษร

ประกอบด้วย

COMMAND	" Alarm List "
	" Control Block Diagram "
	" Control Block Faceplate "
	" Clear System Memory "
	" Digital Block Diagram "
	" Digital Block Faceplate "
	" Edit System Memory "
	" Exit "
	" Graphic Display "

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

“ Graphic Display Generator ”
 “ Group Display ”
 “ Input Block Configuration ”
 “ Initial Screen Memory ”
 “ Network System ”
 “ Output Block Configuration ”
 “ Run mode ”
 “ Read System Memory ”
 “ System Configuration ”
 “ Sequential Control Monitor ”
 “ Sequential Control Program ”
 “ Standby Mode ”
 “ Trend Display ”
 “ Write System Memory ”

2. Alarm_List (AL)

ALARM DISPLAY				
NODE TAG	DESCRIPTOR	ALARM TYPE	DATE	TIME

รูปที่ 4.3 แสดงหน้าจอ Alarm List

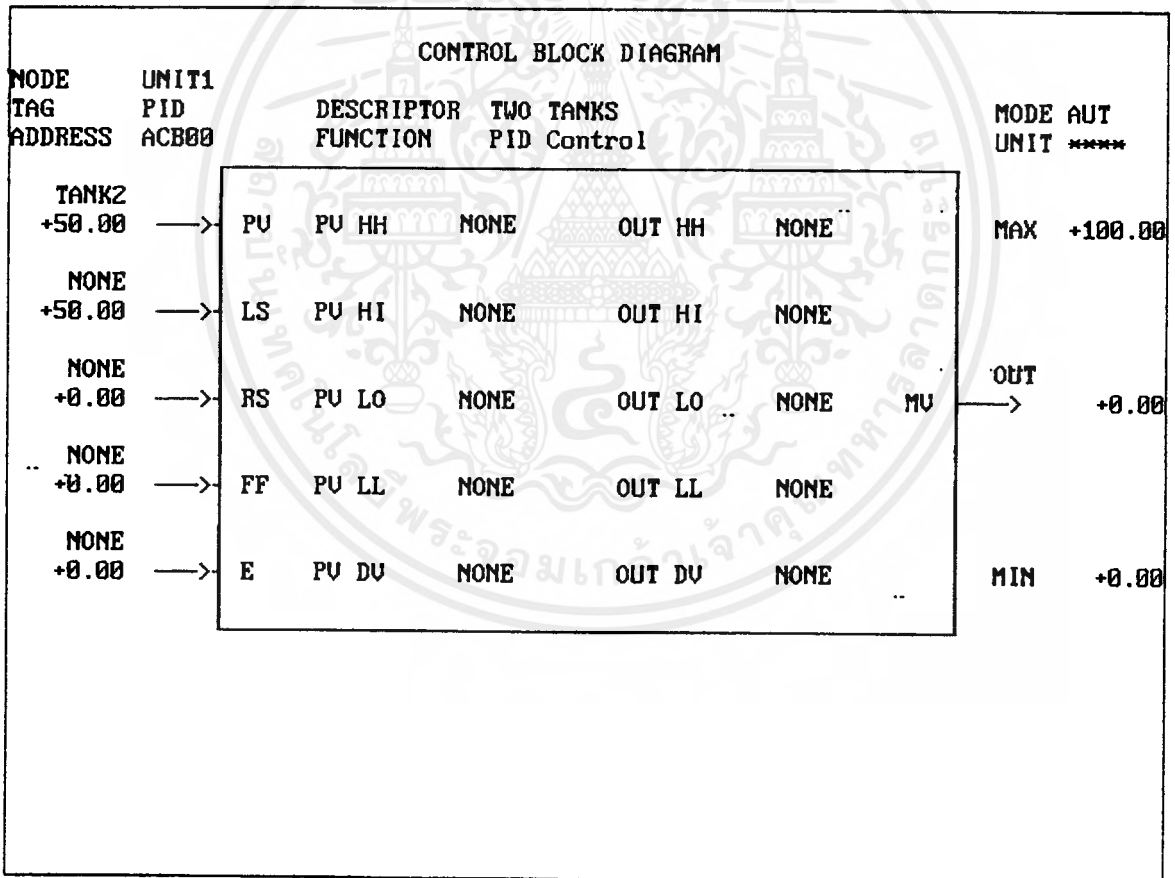
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประโยชน์ อำนวยความสะดวกให้แก่ผู้ใช้ ในการดูการเกิด Alarm ของบล็อกร่างต่าง ๆ ในระบบที่เกิดขึ้น ซึ่งจะช่วยให้ทราบถึง ชนิด , วัน และเวลาของ Alarm ที่เกิดขึ้นได้ ทำให้สามารถแก้ไขและป้องกันระบบได้อย่างทันท่วงที

ประกอบด้วย - ส่วนที่ใส่ชื่อของ Station (NODE) เพื่อใช้ในการค้นหาข้อมูลที่เกิด Alarm ของระบบนั้น ๆ มาแสดง

- ตัวบอกรหัสของบล็อกร่างที่เกิด Alarm (TAG)
- ตัวบรรยายลักษณะระบบของบล็อกร่างที่เกิด Alarm (DESCRIPTION)
- ตัวบอกรหัสของการเกิด Alarm ของแต่ละบล็อกร่าง (ALARM TYPE)
- ตัวบอกรหัสวันที่ของการเกิด Alarm ของแต่ละบล็อกร่าง (DATE)
- ตัวบอกเวลาของการเกิด Alarm ของแต่ละบล็อกร่าง (TIME)

3. Control Block Diagram (CBD)



รูปที่ 4.4 แสดงหน้าจอ Control Block Diagram

ประโยชน์ 1. ช่วยในการ Config ว่าแต่ละบล็อกร่างเป็นอะไรมีฟังก์ชันเป็นอย่างไร

2. ใช้ในการพิจารณา เพื่อช่วยเชื่อมบล็อกร่างต่างๆ เข้าด้วยกันเป็นรูป


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ประกอบด้วย
- ส่วนที่ใช้ชื่อของ Station (NODE)
 - ส่วนที่ใช้ชื่อของบล็อก (TAG)
 - ส่วนที่ใช้คำบรรยายลักษณะของระบบ (DESCRIPTION)
 - ส่วนที่บอกถึงชนิดของการควบคุม (MODE)
 - ส่วนที่บอกตำแหน่งของบล็อก (ADDRESS) ซึ่งมีตั้งแต่ ACB00-ACB99
 - ส่วนที่บอกชนิดของบล็อก (FUNCTION) ซึ่งมีชนิดต่างๆ ดังนี้
 - “ MAN Manual ”
 - “ ON/OFF Control ”
 - “ PID Control ”
 - “ LL Lead/Lag ”
 - “ DT Dead Time ”
 - “ TOT Totalizer ”
 - “ RB Ratio/Bias ”
 - “ SSW Signal Selector Switch ”
 - “ HSW High Selector Switch ”
 - “ LSW Low Selector Switch ”
 - “ PLI Piecewise Interpolation ”
 - “ POLY Polynomial Function ”
 - “ Mass Storage Process ”
 - “ Stirred Tank Heat ”
 - “ Heat Transfer Process ”
 - “ Chemical Reaction Rate ”
 - “ CSTR Mass Balance of A ”
 - “ CSTR Energy Balance ”
 - “ Mixing Process Bal. of A ”
 - “ Mixing Process Energy Bal. ”
 - ส่วนที่บอกหน่วยของระบบ (UNIT) เช่น LITER, TON, Kg เป็นต้น
 - ส่วนที่บอกค่าของตัวแปรที่เข้าบล็อก (INPUT VARIABLE)
 - ส่วนที่บอกค่าของตัวแปรที่ออกจากบล็อก (OUTPUT VARIABLE)
 - ส่วนที่บอกค่าต่ำสุดและค่าสูงสุดของข้อมูล (MIN / MAX)
 - ส่วนที่บอกค่าของตัวแปรที่เซต ALARM ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้.

- “ PV HH “
- “ PV HI “
- “ PV LO “
- “ PV LL “
- “ PV DV “
- “ OUT HH “
- “ OUT HI “
- “ OUT LO “
- “ OUT LL “
- “ OUT DV “

4. Control Block Faceplate (CBF)

CONTROL BLOCK FACEPLATE						MODE AUT
NODE TAG ADDRESS	UNIT1 PID ACB00	DESCRIPTOR FUNCTION	TWO TANKS PID Control			
PU TANK2	+50.00	E	NONE	+0.00		↑PV 50.00
LS NONE	+50.00	F	NONE	+0.00		LS 50.00
RS NONE	+0.00	G	NONE	+0.00		RS 0.000
FF NONE	+0.00	H	NONE	+0.00		FF 0.000
Kp Gain	+100.00					E 0.000
Ti Int Time	+3.00	sec.				F 0.000
Td Dev Time	+1.00	sec.				G 0.000
Kf FF Gain	+0.00					H 0.000
$OUT = Kp*(SP-PV) + \Sigma(Ts/Ti)*(SP-PV) + Kp*(Td/Ts)(\Delta PV) + Kf*FF$						
OUT	+0.00					MV 0.000

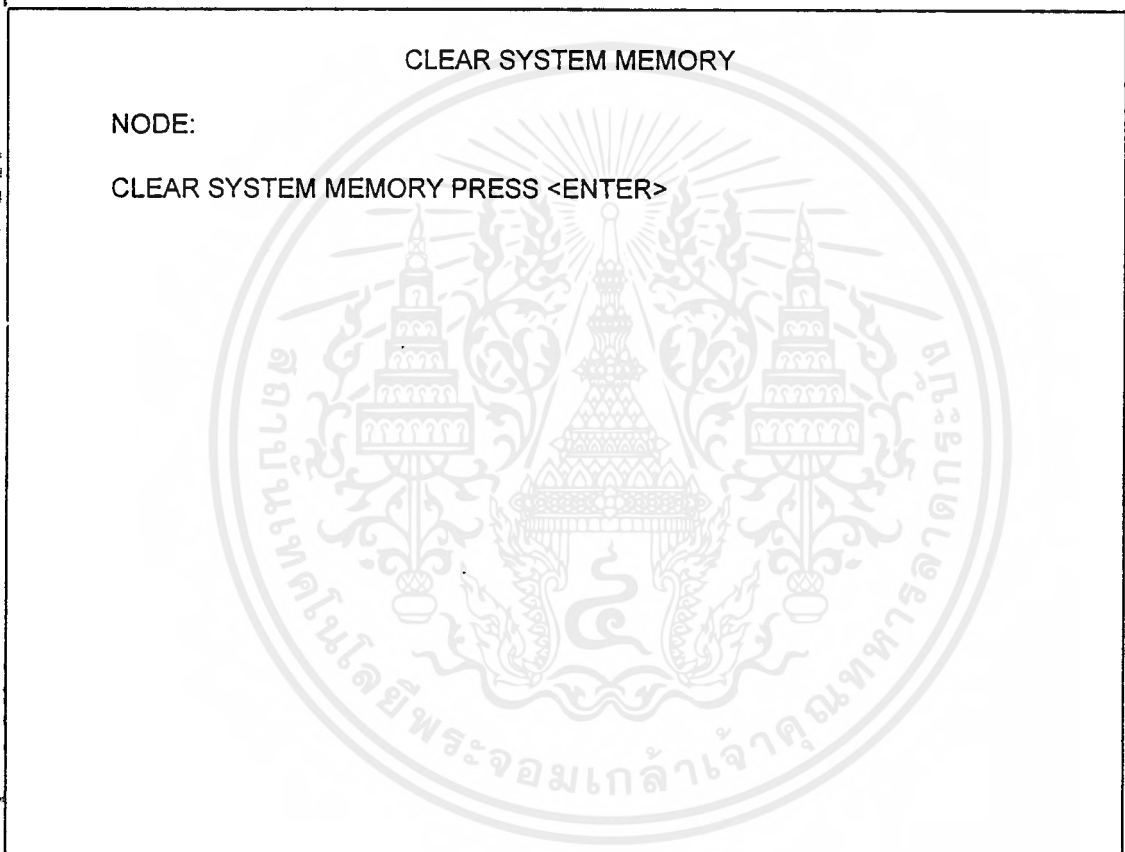
รูปที่ 4.5 แสดงหน้าจอ Control Block Faceplate

ประโยชน์ สำหรับตั้งค่าข้อมูลของบล็อก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ประกอบด้วย
- อินพุทของบล็อก แสดงค่าและเซทค่าของตัวแปรในระบบ
 - ข้อมูลของบล็อกมีไว้เพื่อการเซทค่า ซึ่งจะแตกต่างกันไปตามแต่ละชนิดของฟังก์ชัน
 - สมการของบล็อก เพื่อบอกความสัมพันธ์ของข้อมูลของบล็อก
 - กราฟแท่ง เพื่อแสดงค่าตัวแปรอินพุทและเอาต์พุทของบล็อก

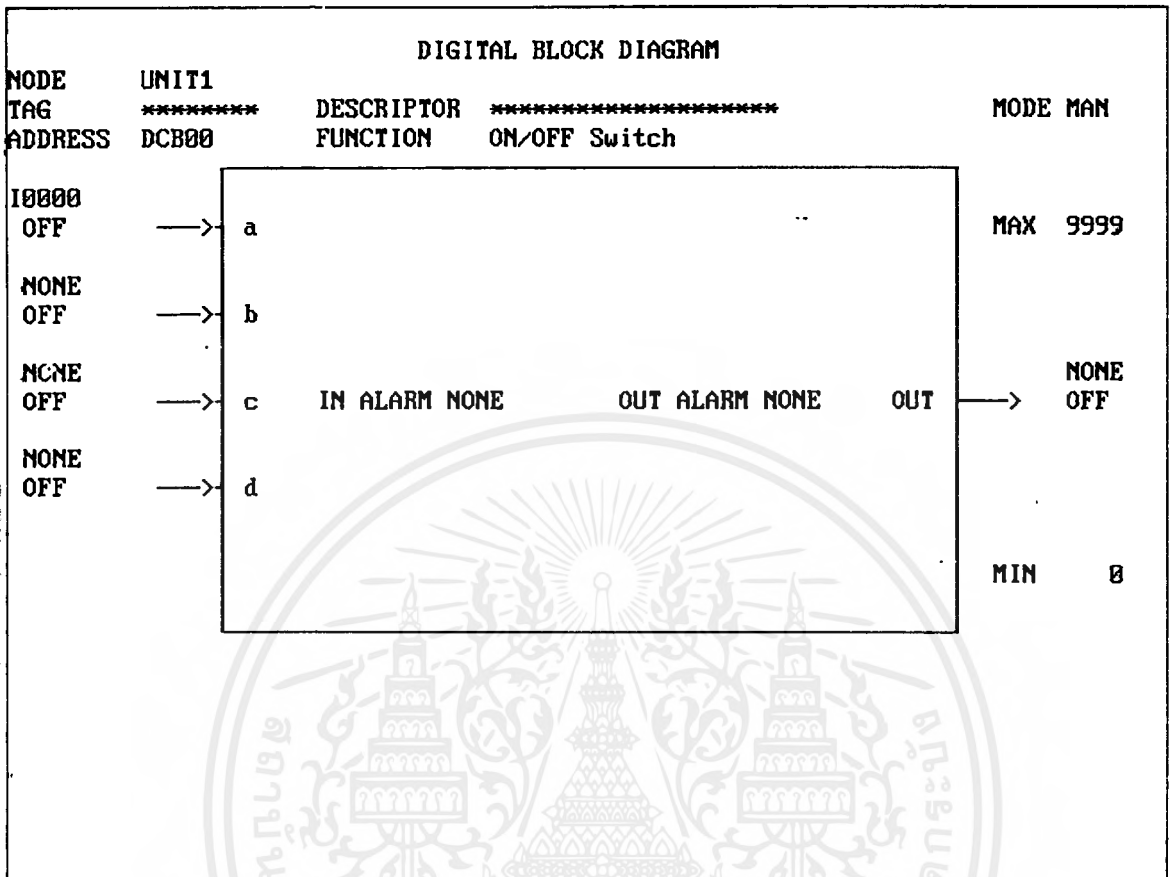
5. Clear System Memory (CSM)



รูปที่ 4.6 แสดงหน้าจอ Clear System Memory

ประโยชน์ ใช้สำหรับลบบล็อกไดอะแกรม ทั้ง ส่วน Control Block, Digital Block, Input Block, Output Block, และ I/O Memory

6. Digital Block Diagram (DBD)



รูปที่ 4.7 แสดงหน้าจอ Digital Block Diagram

- ประโยชน์**
1. ช่วยในการ Config ว่าแต่ละบล็อกเป็นอะไรมีฟังก์ชันเป็นอย่างไร
 2. ใช้ในการพิจารณา เพื่อช่วยเชื่อมบล็อกต่างๆ เข้าด้วยกันเป็นรูป

- ประกอบด้วย**
- ส่วนที่ใส่ชื่อของ Station (NODE)
 - ส่วนที่ใส่ชื่อของบล็อก (TAG)
 - ส่วนที่ใส่คำบรรยายลักษณะของระบบ (DESCRIPTION)
 - ส่วนที่บอกถึงชนิดของการควบคุม (MODE)
 - ส่วนที่บอกตำแหน่งของบล็อก (ADDRESS) ซึ่งมีตั้งแต่ DCB00-DCB99
 - ส่วนที่บอกชนิดของบล็อก (FUNCTION) ซึ่งมีชนิดต่างๆ ดังนี้
 “ ON/OFF Switch ”
 “ Toggle Switch ”

- “ One Shot ”
- “ Shift Register ”
- “ Arithmetics Shift Left ”
- “ Arithmetics Shift Right ”
- “ Move ”
- “ Move Not ”
- “ Compare Less Than ”
- “ Compare Equal ”
- “ Compare Greater Than ”
- “ BCD to Binary ”
- “ Increment ”
- “ Decrement ”
- “ Add ”
- “ Subtract ”
- “ Word AND ”
- “ Word OR ”
- “ Word XOR ”
- “ Cylinder ”
- “ 3-Floor Elevator ”
- “ Conveyor Mechanism ”
- “ Batch Process ”
- ส่วนที่บอกหน่วยของระบบ (UNIT)
- ส่วนที่บอกค่าของตัวแปรที่เข้าบล็อกร (INPUT VARIABLE) ต้องเป็นบิต
- ส่วนที่บอกค่าของตัวแปรที่ออกจากบล็อกร (OUTPUT VARIABLE)
- ส่วนที่บอกค่าต่ำสุดและค่าสูงสุดของข้อมูล (MIN / MAX)
- ส่วนที่บอกค่าของตัวแปรที่เซต ALARM ดังนี้
 - “ IN ALARM “
 - “ OUT ALARM “

7. Digital Block Faceplate (DBF)

DIGITAL BLOCK FACEPLATE							
NODE TAG	UNIT1 ADDRESS	DCB00	DESCRIPTOR FUNCTION	ON/OFF Switch			MODE MAN
a	I0000	OFF	e	NONE	23	0017	a OFF
b	NONE	OFF	f	I00	0	0000	b OFF
c	NONE	OFF	g	NONE	0	0000	c OFF
d	NONE	OFF	h	NONE	0	0000	d OFF
							e 23
							f 0
							g 0
							h 0

OUT = a

OUT OFF

OUT OFF

รูปที่ 4.8 แสดงหน้าจอ Digital Block Faceplate

ประโยชน์

สำหรับตั้งค่าข้อมูลของบล็อก

ประกอบด้วย

- อินพุตที่เป็นดิจิตอลของบล็อก แสดงค่าและเซตค่าของตัวแปรเข้าระบบ
- ข้อมูลของบล็อกมีไว้เพื่อการเซตค่า ซึ่งจะแตกต่างกันไปตามแต่ละชนิดของฟังก์ชัน
- สมการของบล็อก เพื่อบอกความสัมพันธ์ของข้อมูลของบล็อก
- กราฟแท่ง เพื่อแสดงค่าตัวแปรอินพุตและเอาต์พุตของบล็อก

8. Edit System Memory (ESM)

MODE POSITION	UNIT1 DECIMAL	SYSTEM MEMORY				
		HEXADECIMAL	BINARY			INPUT DATA
I00	0	0000	0000	0000	0000	0000
I01	0	0000	0000	0000	0000	0000
I02	0	0000	0000	0000	0000	0000
I03	0	0000	0000	0000	0000	0000
I04	0	0000	0000	0000	0000	0000
I05	0	0000	0000	0000	0000	0000
I06	0	0000	0000	0000	0000	0000
I07	0	0000	0000	0000	0000	0000
I08	0	0000	0000	0000	0000	0000
I09	0	0000	0000	0000	0000	0000
I10	0	0000	0000	0000	0000	0000
I11	0	0000	0000	0000	0000	0000
I12	0	0000	0000	0000	0000	0000
I13	0	0000	0000	0000	0000	0000
I14	0	0000	0000	0000	0000	0000
I15	0	0000	0000	0000	0000	0000
I16	0	0000	0000	0000	0000	0000
I17	0	0000	0000	0000	0000	0000
I18	0	0000	0000	0000	0000	0000
I19	0	0000	0000	0000	0000	0000

รูปที่ 4.9 แสดงหน้าจอ Edit System Memory

ประโยชน์ ใช้สำหรับพิจารณา และเชกค่าอินพุทและเอาต์พุทของระบบผ่านทาง พอร์ตอินพุทเอาต์พุท

ประกอบด้วย

- ตำแหน่ง (Channel) ของพอร์ตอินพุทและเอาต์พุท ซึ่งแต่ละช่อง สัญญาณมี 16 บิต อยู่ 4 แบบ แต่ละแบบมีอย่างละ 64 ตัว คือ I00-I63 Q00-Q63 , F00-F63 และ TC00-TC63
- ค่าข้อมูลชนิดเลขฐานสิบ (Decimal)
- ค่าข้อมูลชนิดเลขฐานสิบหก (Hexadecimal)
- ค่าข้อมูลชนิดเลขฐานสอง (Binary)
- ค่าข้อมูลที่เป็นอินพุท (Input Data) ซึ่งสามารถตรวจสอบและเชกค่าได้

9. Exit (EXIT)

EXIT PROGRAM

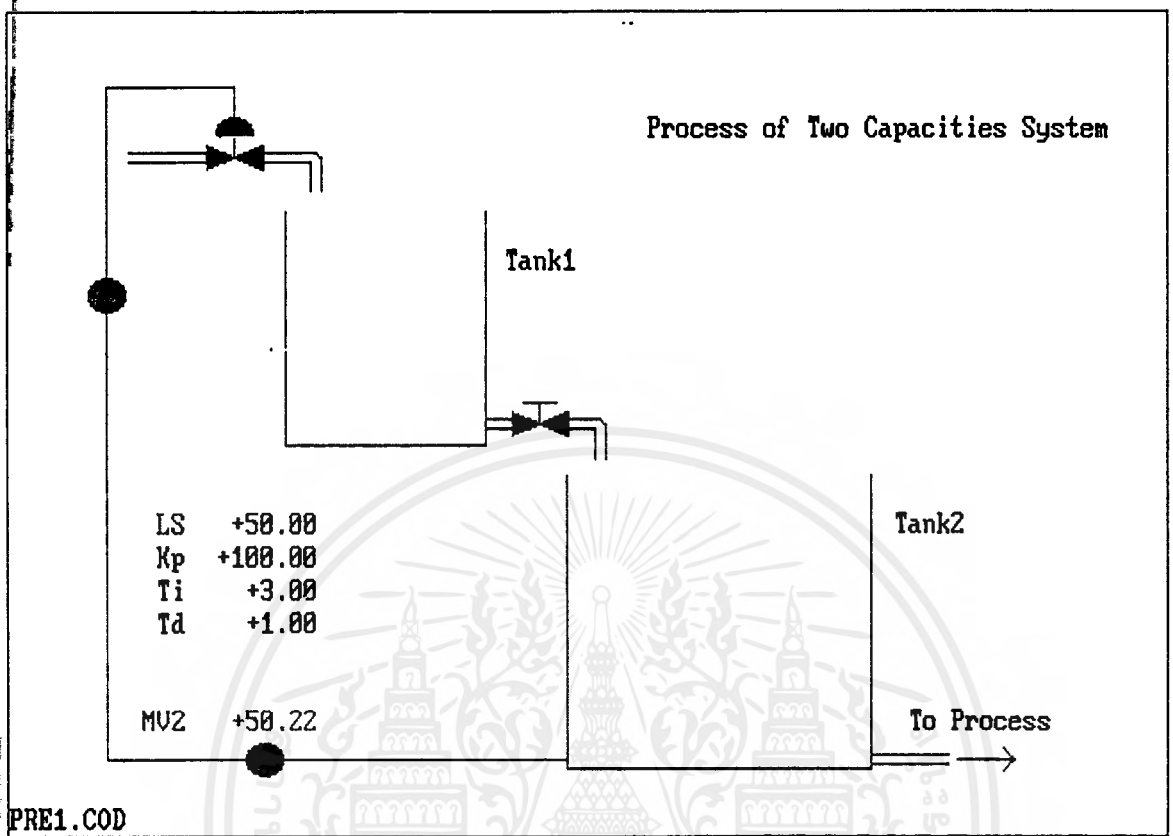
EXIT PROGRAM PRESS <ENTER>

รูปที่ 4.10 แสดงหน้าจอ Exit

ประโยชน์

ใช้เมื่อต้องการออกจากโปรแกรม

10. Graphic Display (GD)



รูปที่ 4.11 แสดงหน้าจอ Graphic Display

ประโยชน์

ใช้แสดงภาพที่วาดในโปรแกรม Gedit และยังสามารถใช้ในการแก้ค่า
ข้อมูลต่างๆ เช่น PV, MV, LS ในขณะที่อยู่ในการทำงานแบบ Run Mode
ได้

11. Graphic Display Generator (GDG)

GRAPHIC CONFIGURATION	
GRAPHIC CONFIGURATION	FILE NAME
	READ
	WRITE
	CLEAR
	EDIT

รูปที่ 4.12 แสดงหน้าจอ Graphic Display Generator

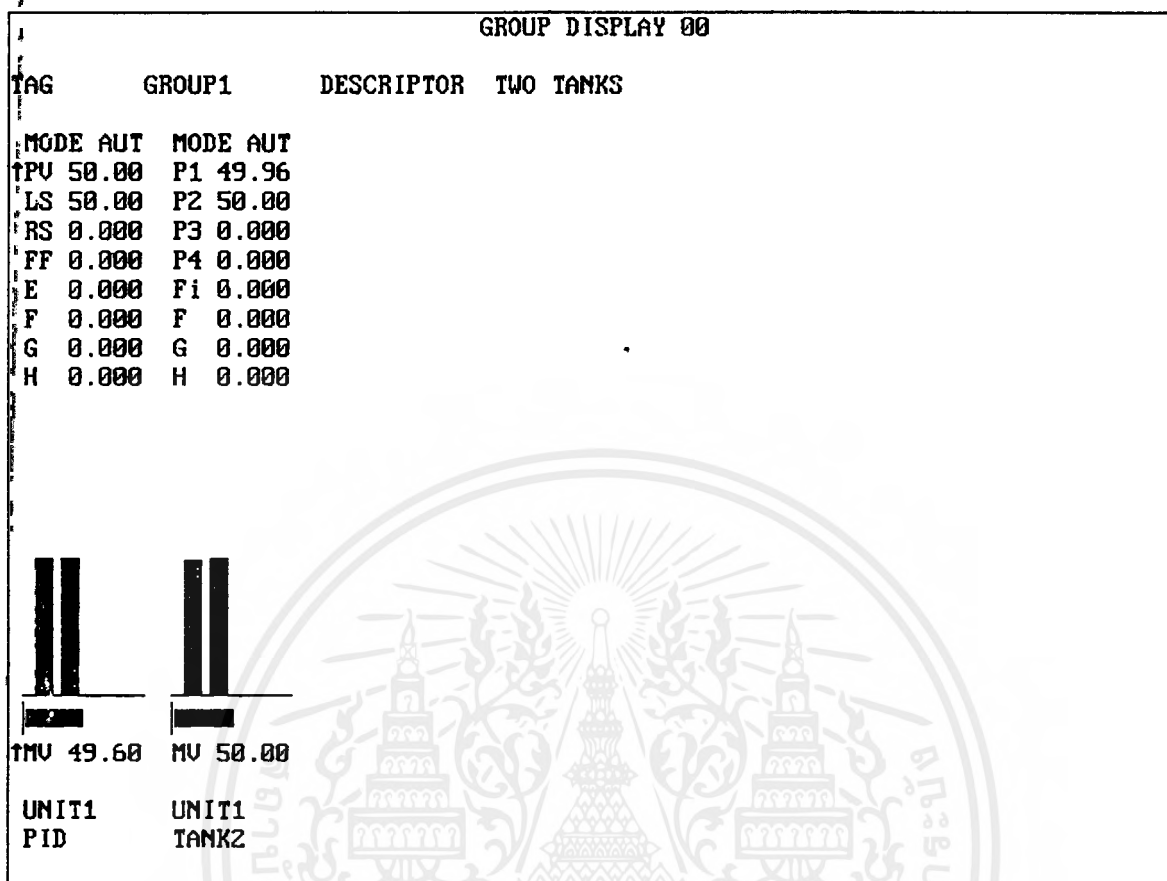
ประโยชน์ ใช้เมื่อต้องการทำการ Config เกี่ยวกับ Graphic Screen ว่าต้องการแสดงในลำดับที่เท่าใดของหน้าจอ

ประกอบด้วย - หน้าจอ Graphic Configuration

มีส่วนประกอบดังนี้

- Read ใช้สำหรับอ่านเพิ่มข้อมูลหน้าจอ Graphic Display ที่ต้องการ
- Write ใช้สำหรับเก็บเพิ่มข้อมูลหน้าจอ Graphic Display ที่ต้องการ
- Clear ใช้สำหรับลบข้อมูลทุกหน้าจอ Graphic Display
- Edit ใช้สำหรับกำหนดลำดับของหน้าจอ Graphic Display ที่ต้องการแสดง

12. Group Display (GD)



รูปที่ 4.13 แสดงหน้าจอ Group Display

ประโยชน์ ใช้สำหรับแสดงค่าและเปลี่ยนค่าข้อมูลได้และในกรณีที่เป็น Process ที่เกี่ยวเนื่องกันสามารถที่จะดูค่าอินพุตเปรียบเทียบกับกันได้ภายในหน้าจอเดียวกัน

ประกอบด้วย

- ส่วนที่ใส่ชื่อของ Station (NODE)
- ส่วนที่ใส่ชื่อของแต่ละหน้าจอของ Group Display (TAG)
- ส่วนที่ใส่คำบรรยายลักษณะของ Group Display (DESCRIPTOR)
- ส่วนที่ใส่ชื่อของ Station และบล็อกซึ่งใน 1 หน้าจอจะแสดงได้ 8 บล็อก ซึ่งเป็นได้ทั้ง Control Block หรือ Digital Block

13. Input Block Configuration (IBC)

INPUT BLOCK CONFIGURATION						
NODE	UNIT1	DESCRIPTOR	*****	MODE	MAN	
TAG	*****		*****	UNIT	*****	
ADDRESS	AIB00					
			OUT HH	NONE		MAX +100.00
			OUT HI	NONE		
NONE	0	IN	OUT LO	NONE	OUT	+0.00
			OUT LL	NONE		
			OUT DU	NONE		MIN +0.00

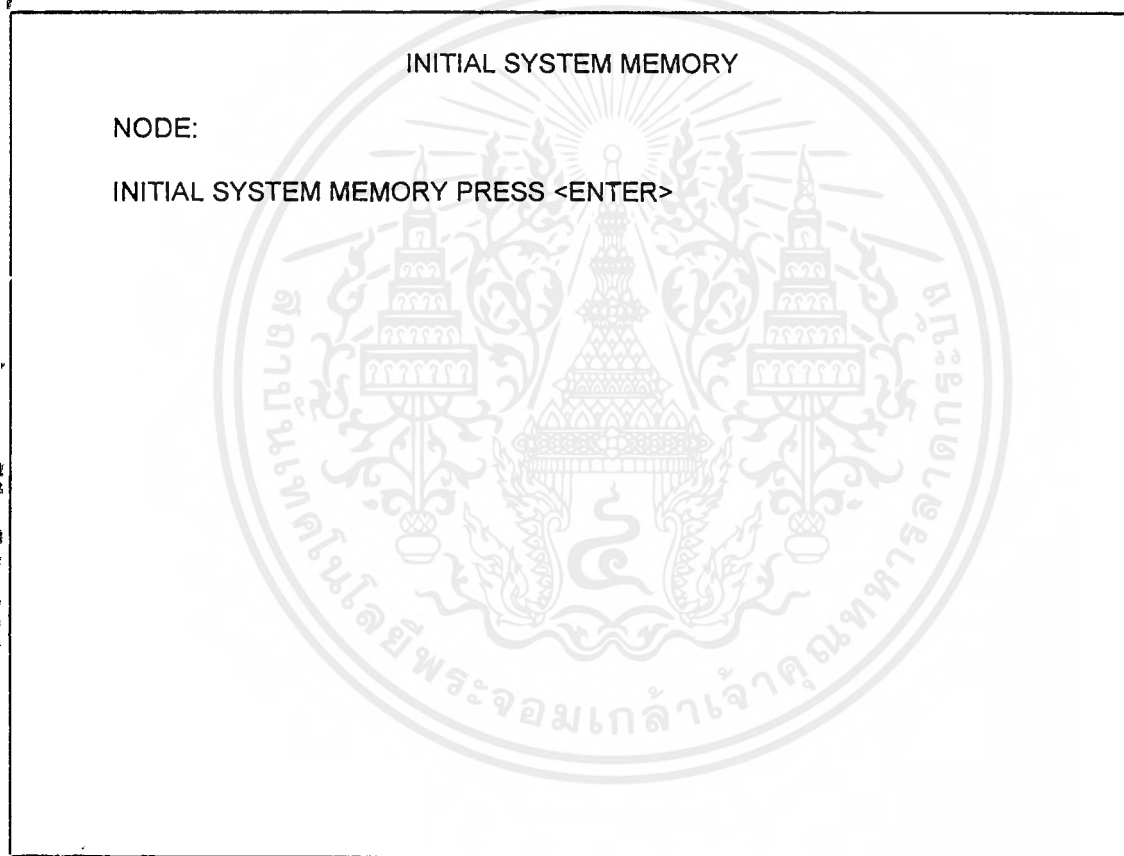
รูปที่ 4.14 แสดงหน้าจอ Input Block Configuration

ประโยชน์ ใช้เป็นตัวเชื่อมอินพุตเมมโมรีที่อยู่ในหน้าจอ Edit System Memory ให้เข้ากับระบบในช่วงที่สามารถกำหนดได้อย่างเหมาะสม

- ประกอบด้วย**
- ส่วนที่ใส่ชื่อของ Station (NODE)
 - ส่วนที่ใส่ชื่อของบล็อก (TAG)
 - ส่วนที่ใส่คำบรรยายลักษณะของระบบ (DESCRIPTION)
 - ส่วนที่บอกถึงชนิดของการควบคุม (MODE)
 - ส่วนที่บอกตำแหน่งของบล็อก (ADDRESS) ซึ่งมีตั้งแต่ AIB00-AIB99
 - ส่วนที่บอกหน่วยของระบบ (UNIT)
 - ส่วนที่บอกค่าของตัวแปรที่เข้าบล็อก (INPUT VARIABLE)
 - ส่วนที่บอกค่าของตัวแปรที่ออกจากบล็อก (OUTPUT VARIABLE)
 - ส่วนที่บอกค่าต่ำสุดและค่าสูงสุดของข้อมูล (MIN / MAX)

- ส่วนที่บอกค่าของตัวแปรที่เซต ALARM ดังนี้
 - “ OUT HH “
 - “ OUT HI “
 - “ OUT LO “
 - “ OUT LL “
 - “ OUT DV “

14. Initial Screen Memory (ISM)



รูปที่ 4.15 แสดงหน้าจอ Initial Screen Memory

ประโยชน์ ใช้เมื่อต้องการลบค่าข้อมูลเฉพาะในส่วนของ I/O Memory

15. Network_System_(NS)

NETWORK SYSTEM			
NO.	NAME	TYPE	DESCRIBE
00	-	-	-
01	UNIT1	CONTROL	-
02	-	-	-
03	UNIT3	OPERATE	-
04	-	-	-
05	-	-	-
06	-	-	-
07	-	-	-
08	-	-	-
09	-	-	-
10	-	-	-

รูปที่ 4.16 แสดงหน้าจอ Network System

ประโยชน์ ใช้เมื่อต้องการดูว่า ณ เวลานั้นมี Node ใดอยู่ในระบบ และเป็น Control Unit หรือ Work Station และสามารถที่จะเขียนบรรยายได้ว่า Node มีลักษณะอย่างไร แต่มีข้อจำกัดที่ต้องดูเฉพาะเมื่อระบบทำงานเป็น Standby Mode เท่านั้น

ประกอบด้วย

- ส่วนที่บอกเลขที่ (NO.) ของ Node
- ส่วนที่บอกชื่อ (NAME) ของ Node
- ส่วนที่บอกชนิด (TYPE) ของ Node (Control หรือ Operate)
- ส่วนที่บอกคำบรรยาย (DESCRIBE) ของ Node

16. Output Block Configuration (OBC)

OUTPUT BLOCK CONFIGURATION					
MODE	UNIT1	DESCRIPTOR	DIRECTION	MODE	MAN
TAG	*****	*****	DIRECT	UNIT	*****
ADDRESS	A0B00				
		IN HH	NONE	MAX	+100.00
		IN HI	NONE		
NONE +0.00	→	IN IN LO	NONE	OUT	→ NONE 0
		IN LL	NONE		
		IN DU	NONE	MIN	+0.00

รูปที่ 4.17 แสดงหน้าจอ Output Block Configuration

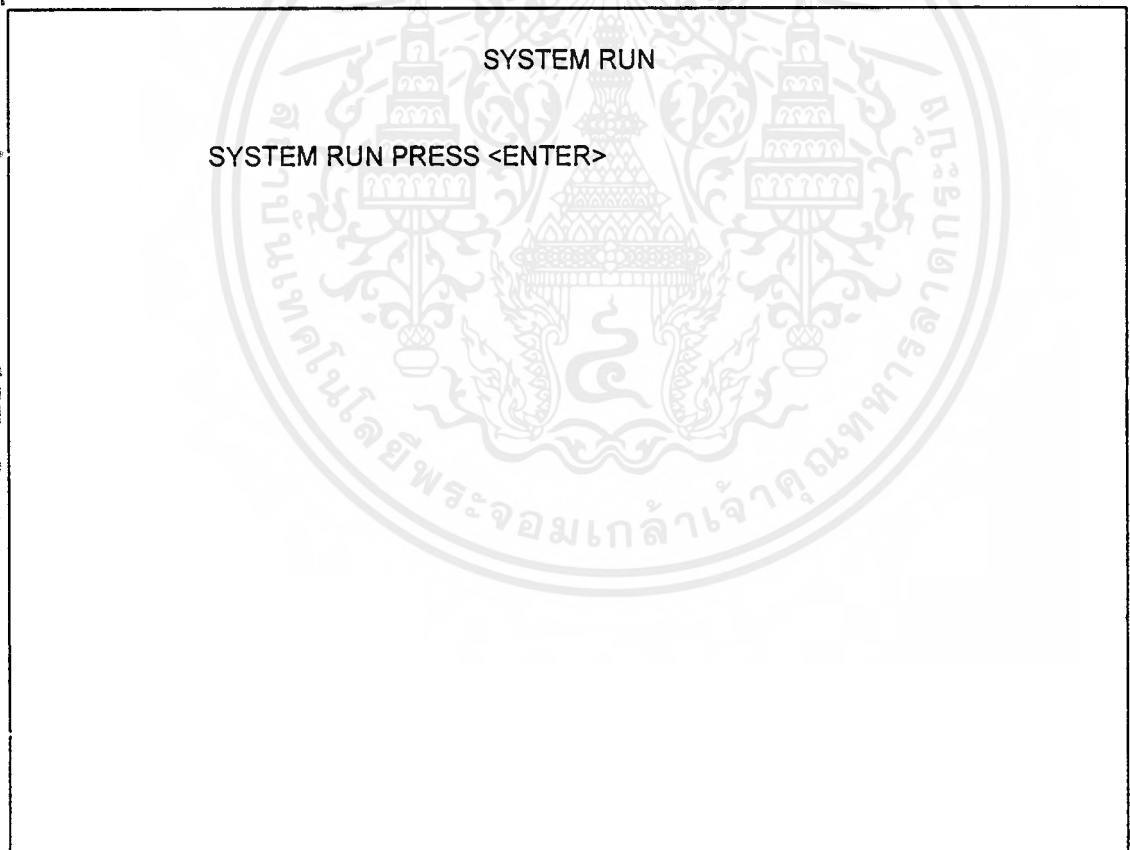
ประโยชน์ ใช้เป็นตัวเชื่อมเอาท์พุทเมมโมรี่ที่อยู่ในหน้าจอ Edit System Memory ให้เข้ากับระบบในช่วงที่สามารถกำหนดได้อย่างเหมาะสม

- ประกอบด้วย**
- ส่วนที่ใส่ชื่อของ Station (NODE)
 - ส่วนที่ใส่ชื่อของบล็อก (TAG)
 - ส่วนที่ใส่คำบรรยายลักษณะของระบบ (DESCRIPTION)
 - ส่วนที่บอกถึงชนิดของการควบคุม (MODE)
 - ส่วนที่บอกตำแหน่งของบล็อก (ADDRESS) ซึ่งมีตั้งแต่ AOB00-AOB99
 - ส่วนที่บอกทิศทางการส่งค่าออกของบล็อกเอาท์พุท (DIRECTION) มีรูปแบบต่างๆ ดังนี้
 - “ DIRECT “
 - “ REVERSE “

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ส่วนที่บอกหน่วยของระบบ (UNIT)
- ส่วนที่บอกค่าของตัวแปรที่เข้าบลิ็อก (INPUT VARIABLE)
- ส่วนที่บอกค่าของตัวแปรที่ออกจากบลิ็อก (OUTPUT VARIABLE)
- ส่วนที่บอกค่าต่ำสุดและค่าสูงสุดของข้อมูล (MIN / MAX)
- ส่วนที่บอกค่าของตัวแปรที่เซต ALARM ดังนี้
 - “ IN HH “
 - “ IN HI “
 - “ IN LO “
 - “ IN LL “
 - “ IN DV “

17. Run Mode (RM)



รูปที่ 4.18 แสดงหน้าจอ Run Mode

ประโยชน์ ใช้เมื่อต้องการให้โปรแกรมทำการควบคุม Process ต่างๆ ในระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

18. Read System Memory (RSM)

READ SYSTEM MEMORY

NODE:

READ SYSTEM MEMORY PRESS <ENTER>

รูปที่ 4.19 แสดงหน้าจอ Read System Memory

ประโยชน์

ใช้เมื่อต้องการอ่านเพิ่มข้อมูลที่เก็บข้อมูล Block Diagram และ I/O Memory ขึ้นมา

19. System Configuration (SC)

SYSTEM CONFIGURATION		
NODE: UNIT1		
SYSTEM CONFIGURATION	FILE NAME	DCS.SYS
SYSTEM MEMORY	FILE NAME	
ALARM LIST	FILE NAME	
STATION CONFIGURATION	FILE NAME	WORK.SYS
GRAPHIC CONFIGURATION	FILE NAME	
GROUP CONFIGURATION	FILE NAME	
TREND CONFIGURATION	FILE NAME	
SEQUENTIAL PROGRAM	FILE NAME	

รูปที่ 4.20 แสดงหน้าจอ System Configuration

- ประโยชน์** ใช้เมื่อต้องการบันทึกค่า Config ต่างๆของระบบ และยังสามารถตั้งค่าสถานะของระบบได้อีกด้วย เช่น ตั้ง Alarm List ให้ On/Off ได้เป็นต้น
- ประกอบด้วย**
- ส่วนที่ใช้ชื่อของ Station (NODE)
 - ส่วนที่ใช้ตั้งชื่อเพิ่มข้อมูล Configuration ของ Control Unit (System Configuration) และใช้สำหรับเก็บชื่อเพิ่มข้อมูล Memory Configuration และ Alarm List มีส่วนประกอบดังนี้
 - Read ใช้สำหรับอ่านเพิ่มข้อมูล System Configuration ที่ต้องการ
 - Write ใช้สำหรับเก็บเพิ่มข้อมูล System Configuration ที่ต้องการ
 - ส่วนที่ใช้ตั้งชื่อเพิ่มข้อมูล Memory Configuration ของ Control Unit (System Configuration) มีส่วนประกอบดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Read ใช้สำหรับอ่านเพิ่มข้อมูลหน้าจอที่ต้องการ
- Write ใช้สำหรับเก็บเพิ่มข้อมูลหน้าจอที่ต้องการ
- Clear ใช้สำหรับลบข้อมูลในเมมโมรี่ของ Control Unit ที่ต้องการ
- ส่วนที่ใช้ตั้งชื่อเพิ่มข้อมูล Alarm ของ Control Unit (Alarm List) และสามารถตั้งค่าสถานะเปิดปิดของ Alarm ได้ โดยมีส่วนประกอบดังนี้
 - Read ใช้สำหรับอ่านเพิ่มข้อมูลหน้าจอ Alarm List ที่ต้องการ
 - Write ใช้สำหรับเก็บเพิ่มข้อมูลหน้าจอ Alarm List ที่ต้องการ
 - Clear ใช้สำหรับลบข้อมูล Alarm ของ Control Unit ที่ต้องการ
 - On ใช้สำหรับทำให้ Alarm List ทำงาน
 - Off ใช้สำหรับทำให้ Alarm List หยุดทำงาน
- ส่วนที่ใช้ตั้งชื่อเพิ่มข้อมูล Configuration ของ Work Station (Station Configuration) และใช้สำหรับเก็บชื่อเพิ่มข้อมูล Graphic - Configuration, Group Configuration, Trend Configuration และ Sequential Program มีส่วนประกอบดังนี้
 - Read ใช้สำหรับอ่านเพิ่มข้อมูล Station Configuration ที่ต้องการ
 - Write ใช้สำหรับเก็บเพิ่มข้อมูล Station Configuration ที่ต้องการ
- ส่วนที่ใช้ตั้งชื่อเพิ่มข้อมูล Graphic Screen ของ Work Station (Graphic Configuration) มีส่วนประกอบดังนี้
 - Read ใช้สำหรับอ่านเพิ่มข้อมูลของหน้าจอ Graphic Display ที่ต้องการ
 - Write ใช้สำหรับเก็บเพิ่มข้อมูลของหน้าจอ Graphic Display ที่ต้องการ
 - Clear ใช้สำหรับลบข้อมูลของทุกหน้าจอ Graphic Display
 - Edit ใช้สำหรับกำหนดลำดับของหน้าจอ Graphic Display ที่ต้องการแสดง

- ส่วนที่ใช้ตั้งชื่อเพิ่มข้อมูล Configuration ของ Group Display (Group Configuration) มีส่วนประกอบดังนี้
 - Read ใช้สำหรับอ่านเพิ่มข้อมูลหน้าจอ Group Display ที่ต้องการ
 - Write ใช้สำหรับเก็บเพิ่มข้อมูลหน้าจอ Group Display ที่ต้องการ
 - Clear ใช้สำหรับลบข้อมูลหน้าจอ Group Display
- ส่วนที่ใช้ตั้งชื่อเพิ่มข้อมูล Trend Display ของ Work Station (Trend Configuration) และสามารถตั้งค่าสถานะเปิดปิดของ Trend ได้โดยมี ส่วนประกอบดังนี้
 - Read ใช้สำหรับอ่านเพิ่มข้อมูลหน้าจอ Trend Display ที่ต้องการ
 - Write ใช้สำหรับเก็บเพิ่มข้อมูลหน้าจอ Trend Display ที่ต้องการ
 - Clear ใช้สำหรับลบข้อมูล Trend Display ของ Work Station ที่ต้องการ
 - On ใช้สำหรับทำให้ Trend Display ทำงาน
 - Off ใช้สำหรับทำให้ Trend Display หยุดทำงาน
- ส่วนที่ใช้ตั้งชื่อเพิ่มข้อมูล Sequential Program ของ Sequential Control Program (Sequential Program) มีส่วนประกอบดังนี้
 - Read ใช้สำหรับอ่านเพิ่มข้อมูลหน้าจอ Sequential Control Program ที่ต้องการ
 - Write ใช้สำหรับเก็บเพิ่มข้อมูลหน้าจอ Sequential Control Program ที่ต้องการ
 - Clear ใช้สำหรับลบข้อมูลหน้าจอ Sequential Control Program

20. Sequential Control Monitor (SCM)

รูปที่ 4.21 แสดงหน้าจอ Sequential Control Monitor

ประโยชน์

ใช้เมื่อต้องการดูการทำงานของโปรแกรม Ladder ที่เขียนแบบการควบคุมแบบ PLC (Programmable Logic Control) ที่เขียนขึ้นใน Sequential Control Program

21. Sequential Control Program (SCP)

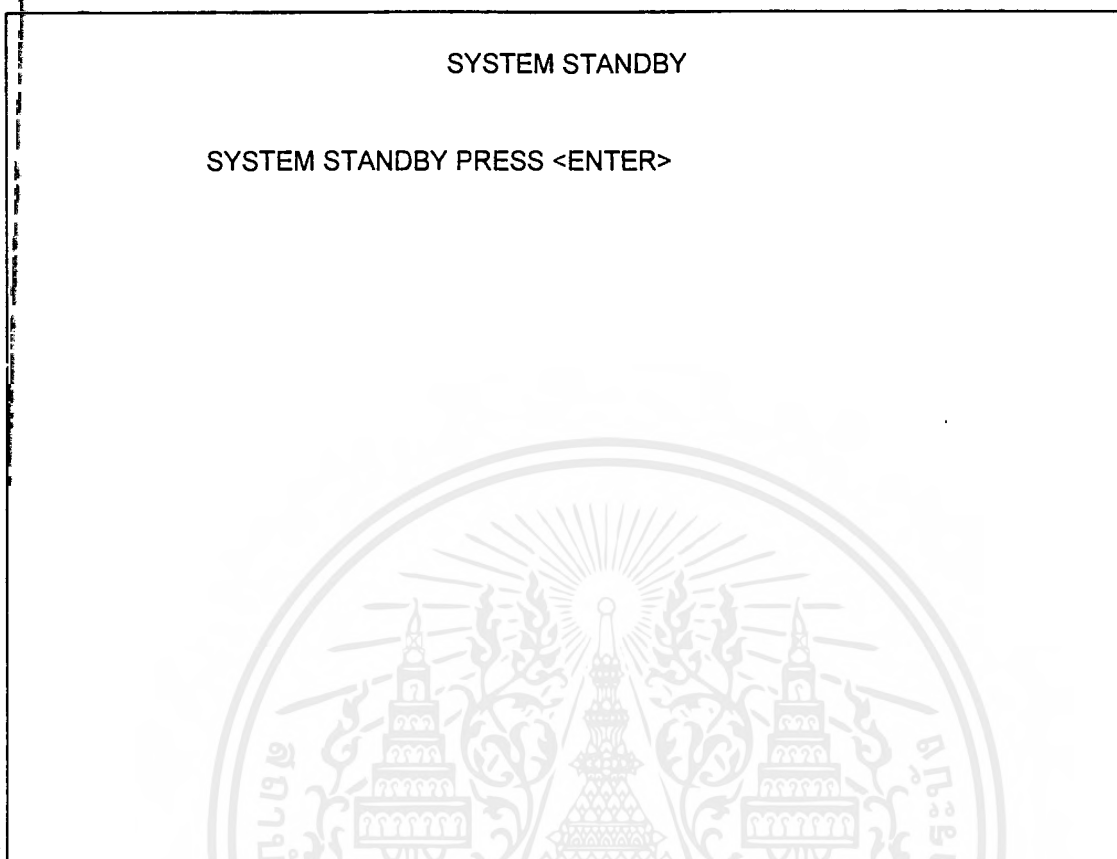


รูปที่ 4.22 แสดงหน้าจอ Sequential Control Program

ประโยชน์

ใช้เมื่อต้องการเขียนโปรแกรม Ladder ที่เลียนแบบการควบคุมแบบ PLC (Programmable Logic Control)

22. Standby Mode (SM)



รูปที่ 4.23 แสดงหน้าจอ Standby Mode

ประโยชน์ ใช้เมื่อต้องการให้โปรแกรมหยุดทำการควบคุม Process ต่างๆ ในระบบ

23. Trend Display (TD)

TREND DISPLAY 00			
TAG	TREND1	DESCRIPTOR	TREND OFF
		TWO TANKS	
		SCAN TIME	1 SEC.
			MAX +100.0
			UNIT1
			TANK2
			•MU +0.00
			UNIT1
			PID
			•MU +0.00
			UNIT1
			PID
			•MU +0.00
			NONE
			NONE
			'
			MIN +0.0

รูปที่ 4.24 แสดงหน้าจอ Trend Display

ประโยชน์ ใช้เมื่อต้องการดูแนวโน้มของข้อมูลต่างในรูปแบบของ Graph เพื่อที่จะนำไปวิเคราะห์การควบคุมของระบบต่อไป

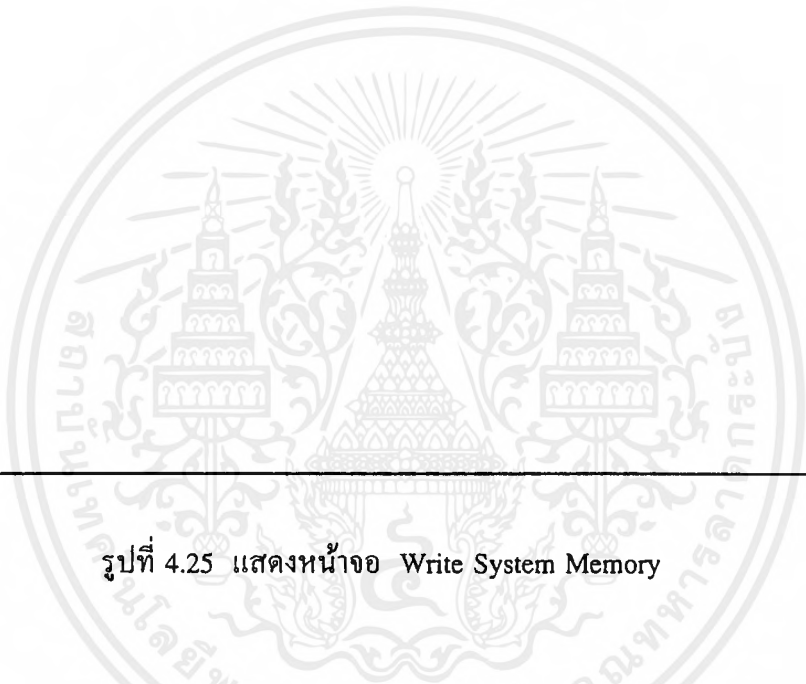
- ประกอบด้วย**
- ส่วนที่ใส่ชื่อของ Station (NODE)
 - ส่วนที่ใส่ชื่อของแต่ละหน้าจอของ Trend Display (TAG)
 - ส่วนที่ใส่คำบรรยายลักษณะของ Trend Display (DESCRIPTOR)
 - ส่วนที่แสดงสถานะการทำงานของ Trend Display (ON/OFF)
 - ส่วนที่ตั้งค่าช่วงเวลาในการตรวจสอบข้อมูล (SCAN TIME)
 - ส่วนที่ใส่ชื่อของ Station และบล็อกซึ่งใน 1 หน้าจอจะแสดงได้ 4 บล็อก ซึ่งเป็นได้ทั้ง Control Block หรือ Digital Block
 - ส่วนที่บอกค่าต่ำสุดและค่าสูงสุดของข้อมูล (MIN / MAX)

24. Write System Memory (WSM)

WRITE SYSTEM MEMORY

NODE:

WRITE SYSTEM MEMORY PRESS <ENTER>



รูปที่ 4.25 แสดงหน้าจอ Write System Memory

ประโยชน์

ใช้เมื่อต้องการเขียนเพิ่มข้อมูลที่เก็บข้อมูล Block Diagram และ I/O Memory ลงไปใน Control Unit

หน้าจอ Control Unit

```

SEND SOLICIT time: 2472 : packet 2381
RECEIVE SOLICIT num:6 packet:2504
SEND FULL DATA
SEND FULL DATA
SEND SOLICIT time: 2511 : packet 2629
RECEIVE SOLICIT num:6 packet:2752
SEND FULL DATA
SEND FULL DATA
SEND SOLICIT time: 2550 : packet 2877
RECEIVE SOLICIT num:6 packet:3000
SEND FULL DATA
SEND FULL DATA
SEND SOLICIT time: 2589 : packet 3125
RECEIVE SOLICIT num:6 packet:3248
SEND FULL DATA
SEND FULL DATA
SEND SOLICIT time: 2628 : packet 3373
RECEIVE SOLICIT num:6 packet:3503
SEND FULL DATA
SEND FULL DATA
SEND SOLICIT time: 2668 : packet 3628
SEND FULL DATA
SEND FULL DATA
RECEIVE SOLICIT num:6 packet:3750

```

รูปที่ 4.26 แสดงหน้าจอ Control Unit

ประโยชน์ ใช้ในการแสดงสถานะการรับส่งข้อมูลในระบบ
ประกอบด้วย

SOCKET TABLE FULL	แสดงสถานะที่ Control Unit ไม่สามารถเปิด Socket ได้
SOCKET ALREADY OPEN	แสดงสถานะที่ Control Unit ไม่สามารถเปิด Socket ได้
INPUT NAME UNIT	ใส่ชื่อ Control Unit
INPUT NUMBER	ใส่หมายเลข Control Unit
IPX NOT INSTALLED	แสดงสถานะที่ Control Unit ไม่ได้ Load Ipx Driver
SEND SOLICIT TIME: 'xx' PACKET: 'xx'	แสดงจำนวน packet ที่ Control Unit ทำการส่ง ภายในช่วง time ที่เปลี่ยนไป
RECEIVE SOLICIT	แสดงสถานะที่ Control Unit ได้รับ Solicit Frame
SEND FULL DATA	แสดงสถานะที่ Control Unit ได้ทำการส่งข้อมูลเพื่อ Update ระบบ
RECEIVE RUN	แสดงสถานะที่ Control Unit ได้รับคำสั่งให้ run ระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

RECEIVE STANDBY	แสดงสถานะที่ Control Unit ได้รับคำสั่งให้ standby ระบบ
RECEIVE REQUEST BY TAG	แสดงสถานะที่ Control Unit ได้รับการร้องขอข้อมูล โดยผ่านการทางการใช้ TAG เป็นตัวอ้างอิง
RECEIVE REQUEST BY BLOCK	แสดงสถานะที่ Control Unit ได้รับการร้องขอข้อมูล โดยผ่านการทางการใช้ BLOCK เป็นตัวอ้างอิง
RECEIVE INITIAL MEMORY	แสดงสถานะที่ Control Unit ได้รับคำสั่งให้ทำการตั้งค่า ให้กับหน่วยความจำ
CLEAR SYSTEM MEMORY	แสดงสถานะที่ Control Unit ได้รับคำสั่งให้ทำการลบ ค่าในหน่วยความจำ
READ FILE	แสดงสถานะที่ Control Unit ได้รับคำสั่งให้ทำการอ่าน เพิ่มข้อมูลที่ต้องการ
WRITE FILE	แสดงสถานะที่ Control Unit ได้รับคำสั่งให้ทำการเขียน เพิ่มข้อมูลที่ต้องการ
RECEIVE SETUP BLOCK	แสดงสถานะที่ Control Unit ได้รับรู้ถึงการเปลี่ยนค่า ต่างๆ ของ BLOCK
RECEIVE SEARCH BLOCK	แสดงสถานะที่ Control Unit ได้รับคำสั่งให้ทำการค้นหา BLOCK
RECEIVE READ ALL FILE	แสดงสถานะที่ Control Unit ได้รับคำสั่งให้ทำการส่ง ชื่อเพิ่มข้อมูลทั้งหมดใน Control Unit

บทที่ 5 ขั้นตอนการทำงานและแบบฝึกหัดการใช้โปรแกรม

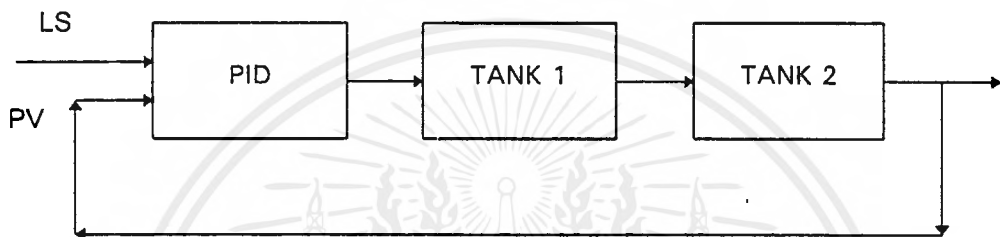
ระบบที่จะใช้เป็นแบบฝึกหัดต่อไปนี้เป็นระบบของถังน้ำที่ถูกทำการควบคุมระดับ โดยใช้ PID Controller ซึ่งมีแผนภาพการควบคุมดังรูปที่ 5.1

ขั้นตอนโดยรวมของการสร้างระบบควบคุม

ขั้นที่ 1 การรีเซทระบบ

ขั้นที่ 2 ทำการสร้างบล็อกต่าง ๆ ขึ้นมา 3 บล็อก คือ PID , Tank 1 , Tank 2

ขั้นที่ 3 นำบล็อกทั้ง 3 มาเชื่อมต่อกันเป็นรูปดังรูป



รูปที่ 5.1 แสดงแผนภาพการควบคุม

ตามหลักที่ว่า นำเอาที่พุดจาก PID บล็อก ไปให้เป็นอินพุตของ Tank 1 Block แล้วนำเอาที่พุดของ Tank 1 Block ไปเป็นอินพุตของ Tank 2 Block สุดท้ายนำเอาที่พุดจาก Tank 2 Block กลับไปเป็นอินพุตของ PID บล็อก

ขั้นที่ 4 การสร้างรูปจากโปรแกรม GEDIT

ขั้นที่ 5 ตั้งค่าเป้าหมายและสร้างส่วนแสดงผล

ส่วนวิธีการใช้โปรแกรมเพื่อการสร้างและการควบคุม เพื่อให้ผู้อ่านได้ทราบ และเข้าใจถึงลักษณะ , วิธีการ และขั้นตอนโดยละเอียดของการใช้โปรแกรม ตามขั้นตอนที่ได้กล่าวมาดังนี้

ขั้นที่ 1 การรีเซทระบบ

1. Run โปรแกรม DCS โดยการเข้าไปที่ `G:/Source/work>work`. แล้วใส่ Input NAME UNIT เป็น "UNIT1" และใส่ Input Number เป็น "1" กด Enter จากนั้นจะเห็นหน้าจอเมนูเลื่อนเมาส์ไปคลิกที่คำว่า Command หรือกด Tab ไปเรื่อย ๆ แทน จนถึงคำว่า " Initial System Memory " แล้วกด Enter อีกครั้ง

หมายเหตุ สามารถใช้คีย์ย่อในการเข้าถึงแต่ละหน้าจอได้ เช่น หน้าจอ STANDBY MODE ใช้ "SM"

2. เลือกไปที่ " Clear System Memory " แล้วกด Enter

ขั้นที่ 2 การสร้างบล็อก และ ขั้นที่ 3 การเชื่อมบล็อก

PID Block

1. กด Tab ไปเรื่อยๆ หรือเลื่อนเมาส์ไปคลิกที่คำว่า Command จนถึงคำว่า " Control - Block Diagram " จะเกิดเป็นคำแทนหน้าจอต่าง ๆ เกิดขึ้น ให้เข้าไปที่หน้าจอต่างๆ

2. หน้าจอ Control Block Diagram

- 2.1 ตั้งค่าแอดเดรส โดยการใช้ Page Up หรือ Page Down ไปที่หน้าจอของแอดเดรส " ACB01 "
- 2.2 ตั้งค่าฟังก์ชันโดยการกด Tab ไปเรื่อย ๆ หรือเลื่อนเมาส์ไปคลิกที่เครื่องหมายดอกจันทน์ แล้วกดปุ่ม Page Up หรือ Page Down ไปที่ฟังก์ชัน " PID Controller "
- 2.3 ตั้งชื่อบล็อก โดยการกด Tab ไปเรื่อย ๆ หรือเลื่อนเมาส์ไปคลิกที่เครื่องหมายดอกจันทน์ หลังคำว่า tag แล้วคีย์คำว่า " PID "
- 2.4 ตั้งค่าบรรยายบล็อก โดยการกด Tab ไปเรื่อย ๆ หรือเลื่อนเมาส์ไปคลิกที่เครื่องหมายดอกจันทน์ หลังคำว่า Descriptor แล้วคีย์คำว่า " Two Tank "
- 2.5 ตั้งชนิดของการควบคุม โดยการกด Tab ไปเรื่อย ๆ หรือเลื่อนเมาส์ไปคลิกที่เครื่องหมายดอกจันทน์ ที่คำว่า Mode แล้วใช้ปุ่ม Page Up หรือ Page Down เปลี่ยนไปที่คำว่า " AUT "
- 2.6 ตั้งค่าอินพุท โดยการกด Tab ไปเรื่อย ๆ หรือเลื่อนเมาส์ไปคลิกที่ " None " แล้วคีย์คำว่า " ACB02 "
- 2.7 ทำการเปลี่ยนหน้าจอไปที่หน้าจอ Control Block Faceplate โดยการกดปุ่ม X หรือ Esc 2 ครั้ง ออกไปที่หน้าจอเริ่มต้น แล้วใช้วิธีการกด Tab ไปเรื่อยๆ หรือเลื่อนเมาส์ไปคลิกที่คำว่า " Control Block Faceplate "

3. หน้าจอ Control Block Faceplate

- 3.1 ตั้งค่า Proportional Gain โดยการกด Tab ไปเรื่อย ๆ หรือเลื่อนเมาส์ไปคลิกจนกระทั่งตัวเลขหลังคำว่า Kp Gain แล้วคีย์ค่า "100.00" ตามด้วยกด Enter
- 3.2 ตั้งค่า Integral Time โดยการกด Tab ไปเรื่อย ๆ หรือเลื่อนเมาส์ไปคลิกที่ตัวเลขหลังคำว่า " Ti Int Time " แล้วคีย์ค่า " 3.00 " กด Enter
- 3.3 ตั้งค่า Derivative Time โดยการกด Tab ไปเรื่อย ๆ หรือเลื่อนเมาส์ไปคลิกที่ตัวเลขหลังคำว่า " Td Dev Time " แล้วคีย์ค่า " 1.00 " กด Enter

Tank 1 Block

1. กด Esc กลับไปที่หน้าจอเมนูเริ่มต้นแล้วคลิกเมาส์ที่ตรง คำว่า " Command " หรือ กด Tab ไปเรื่อย ๆ จนไปที่คำว่า " Control Block Diagram "
2. หน้าจอ Control Block Diagram
 - 2.1 ตั้งค่าแอดเดรส โดยการใช้นุ้ม Page Up หรือ Page Down ไปที่หน้าจอกที่เป็นแอดเดรส " ACB01 "
 - 2.2 ตั้งฟังก์ชันโดยการกด Tab ไปเรื่อย ๆ หรือเลื่อนเมาส์ไปคลิกที่หลังคำว่า " function " แล้วกดปุ่ม Page Up หรือ Page Down ไปเรื่อย ๆ หรือเลื่อนบารไปที่หลังคำว่า " function " แล้วคลิกไปจนถึงที่คำว่า " Mass Storage Process " แล้วกด Enter
 - 2.3 ตั้งชื่อ Block โดยการกด Tab ไปเรื่อย ๆ หรือเลื่อนเมาส์ไปคลิกที่เครื่องหมายดอกจันท์ หลังคำว่า " Tag " หรือเลื่อนเมาส์ไปคลิกที่ เครื่องหมายดอกจันท์ แล้วคีย์คำว่า " Tank 1 " แล้วกด Enter
 - 2.4 ตั้งคำบรรยายบล็อก โดยการกด Tab ไปเรื่อย ๆ หรือเลื่อนเมาส์ไปคลิกที่ เครื่องหมายดอกจันท์ หลังคำว่า "Descriptor" หรือเลื่อนเมาส์ไปคลิกที่ เครื่องหมายดอกจันท์ แล้วคีย์คำว่า " Two Tanks " แล้วกด Enter
 - 2.5 ตั้งชนิดการควบคุมของบล็อก โดยการกด Tab ไปเรื่อย ๆ หรือเลื่อนเมาส์ ไป คลิกที่เครื่องหมายดอกจันท์หลังคำว่า " MODE " จนไปที่ MODE "AUT"
 - 2.6 ตั้งค่า Input ของบล็อก
 - 2.6.1 กด Tab ไปเรื่อย ๆ หรือเลื่อนเมาส์ไปคลิกที่คำว่า " None " แล้วคีย์คำว่า " ACB00 (PID) " กด Enter
 - 2.6.2 กด Tab ไปเรื่อย ๆ หรือเลื่อนเมาส์ไปคลิกที่คำว่า " None " ตัวถัดไปคีย์คำว่า " ACB01 (Tank1 Block) " กด Enter
 - 2.7 ทำการเปลี่ยนหน้าจอไปที่หน้าจอ Control Block Faceplate โดยการกดปุ่ม X หรือ Esc 2 ครั้ง ออกไปที่หน้าจอเริ่มต้น แล้วใช้วิธีการกด Tab ไปเรื่อย ๆ หรือเลื่อนเมาส์ไป คลิกที่คำว่า "Control Block Faceplate"
3. หน้าจอ Control Block Faceplate
 - 3.1 ตั้งค่าขนาดพื้นที่ถัง โดยการกด Tab ไปเรื่อย ๆ หรือเลื่อนเมาส์ไปอยู่หลังคำว่า " Area " แล้วคีย์ค่า " 10.00 " กด Enter
 - 3.2 ตั้งค่าความต้านทานขาเข้าของถัง โดยการกด Tab ไปเรื่อย ๆ หรือเลื่อนเมาส์ ไปคลิกหลังคำว่า " Rin แล้วคีย์ค่า " 10.00 " กด Enter

3.3 ตั้งค่าความต้านทานขาออกของถัง โดยการกด Tab ไปเรื่อย ๆ หรือเลื่อนเมาส์ไปคลิกหลังคำว่า " Rout " แล้วคีย์ค่า " 10.00 " กด Enter

Tank 2 Block จะเหมือนกับ Tank 1 Block

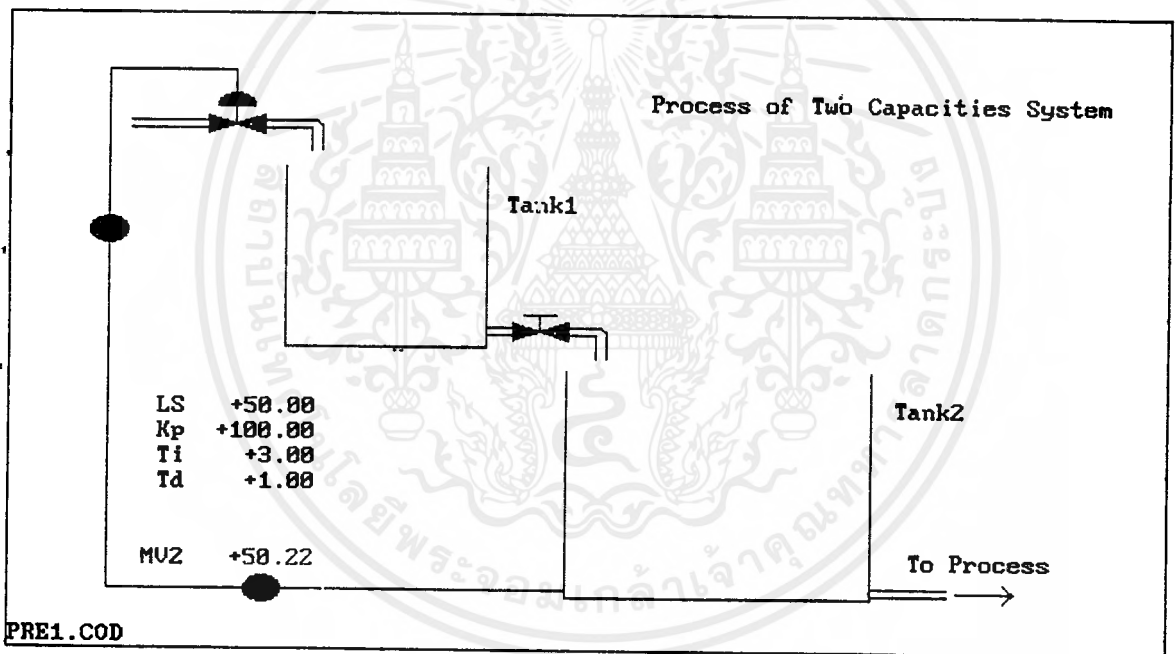
1. กด Esc กลับไปที่หน้าจอเมนูเริ่มต้นแล้วคลิกเมาส์ที่ตรง คำว่า " Command " หรือ กด Tab ไปเรื่อย ๆ จนไปที่คำว่า " Control Block Diagram "
2. หน้าจอ Control Block Diagram
 - 2.1 ตั้งค่าแอดเดรส โดยการใช้นุ่ม Page Up หรือ Page Down ไปที่หน้าจอที่เป็นแอดเดรส " ACB01 "
 - 2.2 ตั้งฟังก์ชัน โดยการกด Tab ไปเรื่อย ๆ หรือเลื่อนเมาส์ไปคลิกที่หลังคำว่า " function " แล้วกดปุ่ม Page Up หรือ Page Down ไปเรื่อย ๆ หรือเลื่อนบาร์ ไปที่หลังคำว่า " function " แล้วคลิกไปจนถึงที่คำว่า " Mass Storage Process " แล้วกด Enter
 - 2.3 ตั้งชื่อ Block โดยการกด Tab ไปเรื่อย ๆ หรือเลื่อนเมาส์ไปคลิกที่เครื่องหมายดอกจันทน์ หลังคำว่า " Tag " หรือเลื่อนเมาส์ไปคลิกที่ เครื่องหมายดอกจันทน์ แล้วคีย์คำว่า " Tank 2 " แล้วกด Enter
 - 2.4 ตั้งคำบรรยายบล็อก โดยการกด Tab ไปเรื่อย ๆ หรือเลื่อนเมาส์ไปคลิกที่เครื่องหมายดอกจันทน์ หลังคำว่า "Descriptor" หรือเลื่อนเมาส์ไปคลิกที่เครื่องหมายดอกจันทน์ แล้วคีย์คำว่า " Two Tanks " แล้วกด Enter
 - 2.5 ตั้งชนิดการควบคุมของบล็อก โดยการกด Tab ไปเรื่อย ๆ หรือเลื่อนเมาส์ไปคลิกที่เครื่องหมายดอกจันทน์หลังคำว่า "MODE" จนไปที่ MODE " AUT"
 - 2.6 ตั้งค่า Input ของบล็อก
 - 2.6.1 กด Tab ไปเรื่อย ๆ หรือเลื่อนเมาส์ไปคลิกที่คำว่า " None " แล้วคีย์ คำว่า " ACB01 (Tank 1) " กด Enter
 - 2.6.2 กด Tab ไปเรื่อย ๆ หรือเลื่อนเมาส์ไปคลิกที่คำว่า " None " ตัวถัดไปคีย์คำว่า " ACB02 (Tank2 Block) " กด Enter
 - 2.7 ทำการเปลี่ยนหน้าจอไปที่หน้าจอ Control Block Faceplate โดยการกดปุ่ม X หรือ Esc 2 ครั้ง ออกไปที่หน้าจอเริ่มต้น แล้วใช้วิธีการกด Tab ไป เรื่อย ๆ หรือเลื่อนเมาส์ไปคลิกที่คำว่า " Control Block Faceplate "

3. หน้าจอ Control Block Faceplate

- 3.1 ตั้งค่าขนาดพื้นที่ถัง โดยการกด Tab ไปเรื่อย ๆ หรือเลื่อนเมาส์ไปอยู่
หลังคำว่า " Area " แล้วคีย์ค่า " 10.00 " กด Enter
- 3.2 ตั้งค่าความต้านทานขาเข้าของถัง โดยการกด Tab ไปเรื่อย ๆ หรือเลื่อน
เมาส์ไปคลิกหลังคำว่า " Rin แล้วคีย์ค่า " 10.00 " กด Enter
- 3.3 ตั้งค่าความต้านทานขาออกของถัง โดยการกด Tab ไปเรื่อย ๆ หรือเลื่อน
เมาส์ไปคลิกหลังคำว่า " Rout " แล้วคีย์ค่า " 10.00 " กด Enter

ขั้นที่ 4 การวาดรูป Process

Process ที่จะใช้เป็นแบบฝึกหัดในการสร้างรูปมีลักษณะดังต่อไปนี้



รูปที่ 5.2 แสดง Plant

มีขั้นตอนดังต่อไปนี้

1. Run โปรแกรม DCS โดยการเข้าไปที่ G:/Source/work>graph แล้วใส่ Input
NAME UNIT เป็น UNIT และใส่ Input Number เป็น "1" กดปุ่ม Enter
2. หลังจากขั้นตอนที่ 1 หน้าจอเปลี่ยนเป็นหน้าจอ " Graphic Display Generator " ดัง
รูปที่ 4.12

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 2.1 ตั้งชื่อรูป โดยการกด Tab ไปเรื่อย ๆ หรือเลื่อนเมาส์ไปคลิกหลังคำว่า " FILENAME " แล้วใส่คำว่า " Pre1.cod " กด Enter
- 2.2 กด Tab ต่อไปเรื่อย ๆ หรือ เลื่อนเมาส์ไปคลิกหลังคำว่า " EDIT " แล้วกด Enter
- 2.3 การเลือกชนิดของหน้าจอ โดยการกด Tab ไปเรื่อย ๆ หรือเลื่อนเมาส์ไปคลิกจนกระทั่งถึงหน้าจอที่มีลักษณะดังนี้

Video Mode.ERESCOLOR

X pixels : 640

Y pixels : 350

Columns : 80

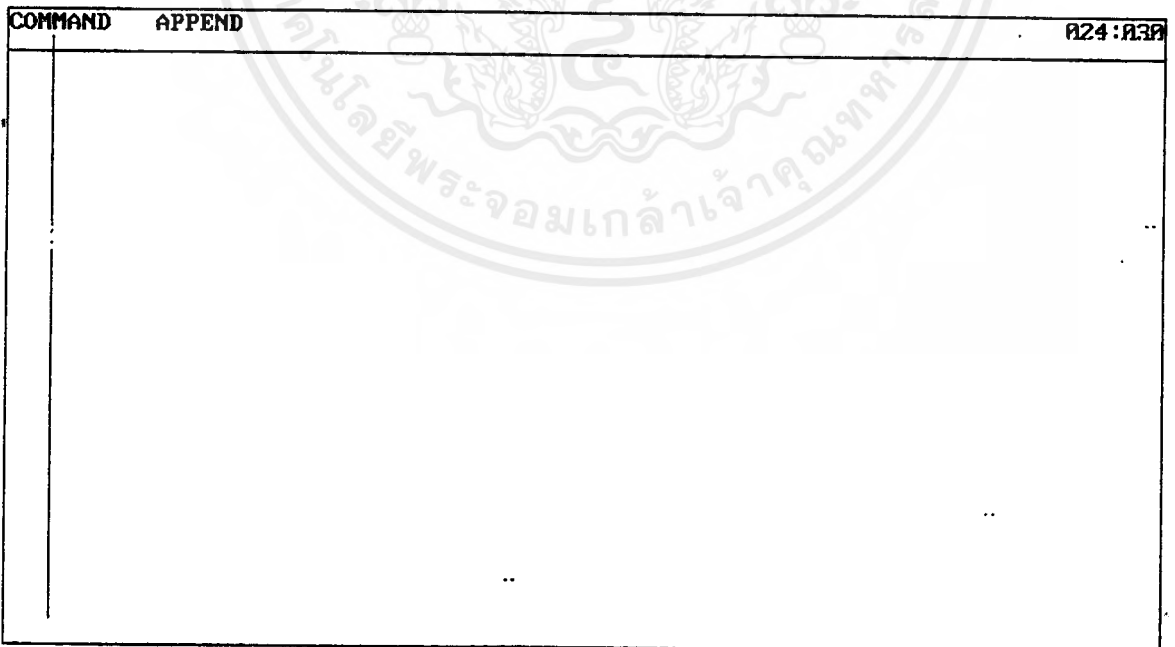
Rows : 25

Color : 16

Pages : 2

Memory : 256

- 2.4 หลังจากขั้นตอนที่ 2.3 หน้าจอจะเปลี่ยนเป็นหน้าจอ GEDIT ดังรูป



รูปที่ 5.3 แสดงหน้าจอ GEDIT

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 2.4.1 กด Tab จนกระทั่งขึ้นคำว่า " APPEND " กดปุ่ม Enter
- 2.4.2 กด Tab จนกระทั่งขึ้นคำว่า " LINE " กดปุ่ม Enter
- 2.4.3 ทำการสร้างรูปถึง 2 ถึง โดยการ
- เลือกจุดเริ่มต้นของเส้นโดยใช้ปุ่มลูกศรเลือกทิศทางการที่ต้องการตามตำแหน่งของถังในรูป
 - เลือก End Point แล้วกด Enter
 - เลือก Color โดยการกด Tab ให้เป็น "WHITE" แล้วกดปุ่ม Enter ให้มีลักษณะเป็นดังรูป
- 2.4.4 ทำการสร้างระดับน้ำขึ้นมาในถัง โดยการกด Esc ออกไปที่ "APPEND" แล้วกด Tab ไปเรื่อย ๆ หรือเลื่อนเมาส์ไปคลิกที่คำว่า "BAR" หลังจากนั้นกด Enter เพื่อ
- เลือก First Corner แล้วกด Enter
 - เลือก Second Corner แล้วกด Enter
 - เลือก Color โดยการกด Tab ให้เป็น " BLUE " แล้วกด Enter ให้มีลักษณะออกมาเป็นดังรูป
- 2.4.5 ทำการสร้าง Value โดยกด Esc ออกไปที่ " APPEND " แล้วกด Tab ไปที่คำว่า " SYMBOL " กด Enter
- กด Tab ไปเรื่อย ๆ หรือเลื่อนเมาส์ไปคลิกที่หลังคำว่า "SYMBOL" เปลี่ยนไปเลือกที่คำว่า " Value " กด Enter
 - เลือกขนาดของ Value โดยการกด Tab เลือกไปที่ Size " 9 " กด Enter
 - เลือกทิศทางของ Value โดยการกด Tab เลือกไปที่ "180 DEGREE " กด Enter
 - เลือกตำแหน่งโดยการใช้นิ้วลูกศร ไปที่ตำแหน่งที่ต้องการดังรูปแล้ว กด Enter
 - เลือกสีของ Value โดยการกด Tab หรือปุ่ม Shift Tab เลือกไปที่คำว่า " GRAY "
 - เลือกสีขอบ Value โดยการกด Tab หรือปุ่ม Shift Tab เลือกไปที่คำว่า " WHITE "
- 2.4.6 ทำการสร้างท่อ โดยกด Esc ออกไปที่ " APPEND " แล้วกด Tab ไปที่คำว่า " PIPE "

- เลือกตำแหน่งจุดเริ่มต้นของท่อ " START PIPE " โดยการคลิกจุดศูนย์กลางของท่อ แล้วกด Enter
- เลือกตำแหน่งจุดสิ้นสุดของท่อ " END PIPE " โดยการคลิกจุดศูนย์กลางของท่อ แล้วกด Enter
- เลือกขนาดของท่อ โดยกด Tab เลือกไปที่ Size " 7 "
- เลือกสีของท่อ โดยกด Tab เลือกไปที่ Color " WHITE "

2.4.7 ทำการสร้างตัวแปลงและส่งสัญญาณไฟฟ้า โดยการกด Esc ออกไปที่ "APPEND" แล้วกด Tab ไปเรื่อย ๆ หรือเลื่อนเมาส์ไปคลิกที่คำว่า " Transmitter " กด Enter

- เลือกขนาดของ Transmitter โดยการกด Tab หรือ Shift Tab เลือกไปที่ Size " 10 " กด Enter
- เลือกตำแหน่ง (Position) โดยการคลิกจุดศูนย์กลางของท่อ ให้ได้ตำแหน่งที่ต้องการตามรูป กด Enter
- เลือกสีของ Transmitter โดยการกด Tab โดยการคลิกจุดศูนย์กลาง เลือกไปที่ Color " GRAY " กด Enter
- เลือกสีขอบ โดยการกด Tab โดยการคลิกจุดศูนย์กลาง เลือกไปที่ EDGE " WHITE " กด Enter

2.4.8 ทำการสร้างตัวควบคุม โดยการกด Esc ออกไปที่ " APPEND " แล้วกด Tab ไปเรื่อย ๆ หรือเลื่อนเมาส์ไปคลิกที่คำว่า "Controller" กด Enter

- เลือกขนาดของ Controller โดยการกด Tab หรือ Shift Tab เลือกไปที่ Size " 10 " กด Enter
- เลือกตำแหน่ง (Position) โดยการคลิกจุดศูนย์กลางของท่อ ให้ได้ตำแหน่งที่ต้องการตามรูป กด Enter
- เลือกสีของ Transmitter โดยการกด Tab โดยการคลิกจุดศูนย์กลาง เลือกไปที่ Color " CYAN " กด Enter
- เลือกสีขอบ โดยการกด Tab โดยการคลิกจุดศูนย์กลาง เลือกไปที่ EDGE " WHITE " กด Enter

2.4.9 ทำการสร้างเส้นตรงมาเชื่อมอุปกรณ์ต่าง ๆ เข้าด้วยกัน รวมทั้งสร้างรูปลูกศร โดยใช้วิธีการการสร้างเดียวกันกับการสร้างถึงหน้า 2 ถึง ในหัวข้อ 2.3

2.4.10 ทำการสร้างตัวอักษร โดยการกด Esc ออกไปที่ "APPEND" แล้วกด Tab ไปเรื่อย ๆ หรือเลื่อนเมาส์ไปคลิกที่คำว่า "TEXT" กดปุ่ม Enter

- ใส่ตัวอักษรคำว่า " LS " กด Enter
- เลือกตำแหน่งของตัวอักษร โดยการกดปุ่มลูกศรไปที่ตำแหน่งที่ต้องการตามรูป แล้วกด Enter
- เลือกสีของ Transmitter โดยการกด Tab ไปเรื่อย ๆ หรือเลื่อนเมาส์ไปคลิก โดยการใช้ปุ่มลูกศร เลือกไปที่ Color " YELLOW " กด Enter

กระทำซ้ำในลักษณะเดียวกันกับ 3 ขั้นตอนที่ผ่านมา โดยเปลี่ยนตัวอักษรเป็น " Kp ", " Ti ", " Td ", " MV ", " Process of Two Capacities system ", " Tank 1 ", " Tank 2 " และ " To Process " และเลือกสีให้กับอักษร 3 คำแรกเป็น " WHITE ", " MV " เป็น " LIGHT GREEN " ส่วน " Process of Two Capacities system " เป็น " LIGHT RED " และ 3 ตัวสุดท้ายเป็น " GRAY "

2.4.11 ทำการกำหนดชนิดของตัวอักษรที่สร้างขึ้นมา

ก. ชนิดของตัวอักษรเป็นตัวแปร กด Esc ออกไปที่ "APPEND" แล้วกด Tab ไปที่คำว่า " VARIABLE " กด Enter

- เลือกตำแหน่ง (Position) โดยใช้ปุ่มลูกศรไปที่ตำแหน่งที่ต้องการตามรูป หลังตัวอักษร LS และ MV กด Enter
- เลือกสีของตัวแปร โดยการกด Tab ไปเรื่อย ๆ หรือเลื่อนเมาส์ไปคลิกที่ " COLOR " เป็น " YELLOW " สำหรับตัวแปรที่อยู่หลัง LS " LIGHT GREEN " สำหรับตัวแปรที่อยู่หลัง MV
- ตั้งค่า NODE โดยคีย์คำว่า " UNIT1 " กด Enter

ข. ชนิดของข้อมูลเป็นตัวแปร กด Esc ออกไปที่ " APPEND " แล้วกด Tab ไปที่คำว่า " DATA " กด Enter

- เลือกตำแหน่ง (Position) โดยใช้ปุ่มลูกศรไปที่ตำแหน่งที่ต้องการตามรูป
1. หลังตัวอักษร Kp
 2. หลังตัวอักษร Ti

3. หลังตัวอักษร Td

- เลือกสีของข้อมูล โดยการกด Tab ไปเรื่อย ๆ หรือเลื่อนเมาส์ไปคลิกที่ " COLOR " เป็น " WHITE " สำหรับตัวอักษรทั้ง 3 ตัว

- ตั้งค่า NODE โดยคีย์คำว่า "UNIT1" กด Enter

2.4.12 ทำการบันทึกรูปที่สร้างขึ้นโดยการกด Esc 2 ครั้ง กลับไปที่ COMMAND กด Tab ไปเรื่อย ๆ หรือเลื่อนเมาส์ไปคลิกที่คำว่า " SAVE " ใส่ชื่อไฟล์เป็น " TEMP1.QBJ " กด Enter

2.4.13 ทำการออกจากหน้าจอ GEDIT โดยการกด Esc 2 ครั้ง กลับไปที่ COMMAND กด Tab ไปเรื่อย ๆ หรือเลื่อนเมาส์ไปคลิกที่คำว่า " EXIT " กด Enter

2.4.14 หลังจากขั้นตอนที่ 2.4.13 หน้าจอจะเปลี่ยนไปที่หน้าจอ " Graphic Display Generator " กด Tab ไปเรื่อย ๆ หรือเลื่อนเมาส์ไปคลิกที่คำว่า " WRITE " กด Enter

2.4.15 ทำการออกจากหน้าจอ " Graphic Display Generator " กด Tab ไปเรื่อย ๆ หรือเลื่อนเมาส์ไปคลิกที่คำว่า " EXIT " กด Enter

ขั้นที่ 5 การ SET หน้าจอแสดงผล

ส่วน Graphic

1. Run โปรแกรม DCS โดยการเข้าไปที่ G:/Source/work>work. แล้วใส่ Input NAME UNIT เป็น UNIT และใส่ Input Number เป็น "1" กดปุ่ม Enter จากนั้นจะเห็นหน้าจอเมนูเลื่อนเมาส์ไปคลิกที่คำว่า Command หรือกดปุ่ม Tab ไปเรื่อย ๆ แทน จนถึงคำว่า " System - Configuration " หรือ กดคำว่า " SC "

2. ขณะนี้หน้าจอจะเปลี่ยนเป็น หน้าจอ " System Configuration " กด Tab ไปเรื่อย ๆ หรือเลื่อนเมาส์ไปคลิกที่คำว่า " Graphic Configuration " แล้วคีย์คำว่า " GP1.SYS " กด Enter

3. กด Tab ไปเรื่อย ๆ หรือเลื่อนเมาส์ไปคลิกที่คำว่า " EDIT " แล้วคีย์คำว่า " PRE1.COD " กด Enter

4. กด Esc ออกมาที่เมนูหลัก แล้วเข้าไปที่หน้าจอ " System Configuration " แล้วเข้าไปที่ " Graphic Configuration " แล้วกด Tab ไปเรื่อย ๆ หรือเลื่อนเมาส์ไปคลิกที่คำว่า " WRITE "

5. กด Esc ออกมาที่เมนูหลัก แล้วเข้าไปที่หน้าจอ " System Configuration " แล้วเข้าไปที่ " Graphic Configuration " แล้วกด Tab ไปเรื่อย ๆ หรือเลื่อนเมาส์ไปคลิกที่คำว่า " READ "

ส่วนของหน้าจอแสดงผลข้อมูล

6. เข้าไปที่ " System Configuration " กด Enter เลือกที่ " Trend Configuration " คีย์คำว่า " GPI.SYS " กด Enter เลือกไปที่คำว่า " ON " แล้วกด Enter อีกครั้ง

7. เข้าไปที่ " System Configuration " กด Enter เลือกที่ " Alarm List " คีย์คำว่า " GPI.SYS " กด Enter เลือกไปที่คำว่า " ON " แล้วกด Enter อีกครั้ง

8. เข้าไปที่ " System Configuration " กด Enter เลือกที่ " Run Mode " แล้วกด Enter

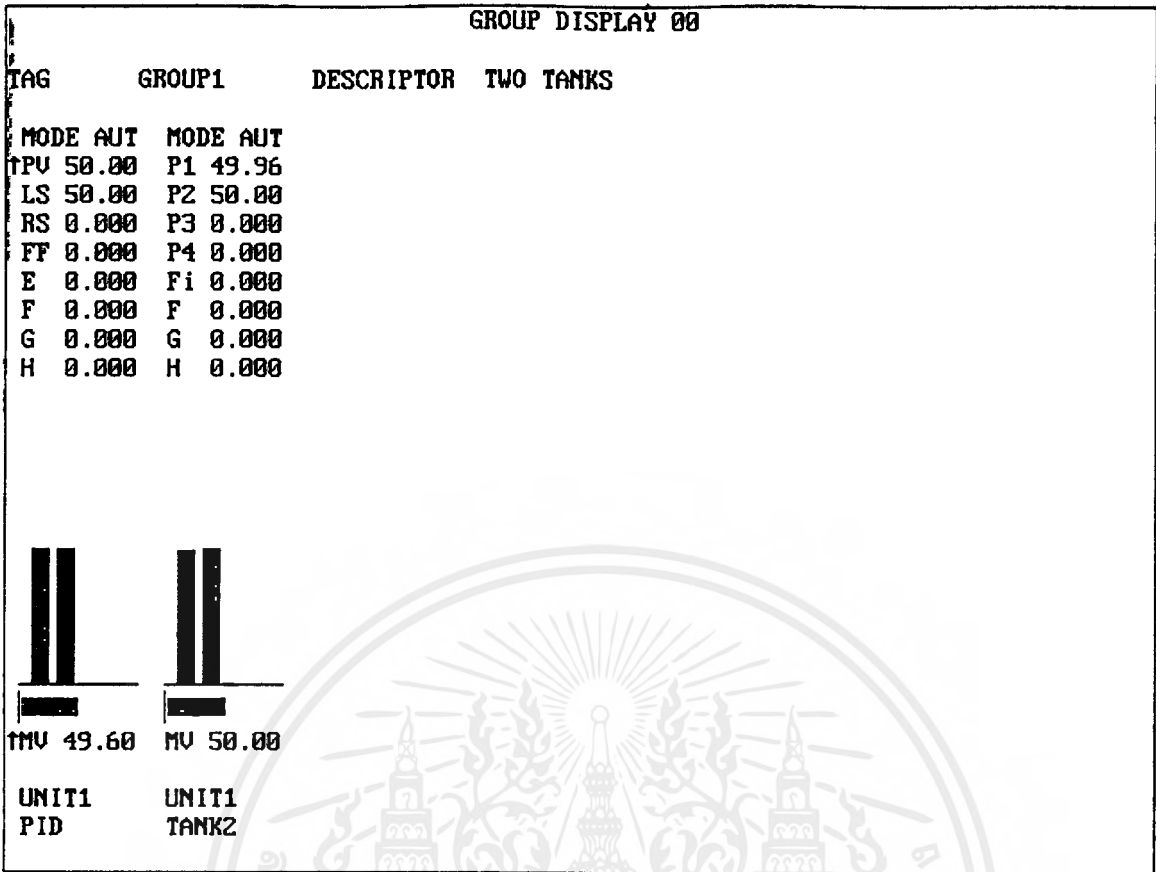
9. เข้าไปที่ " System Configuration " กด Enter เลือกที่ " Graphic Display " กด Enter แล้วสังเกตการเปลี่ยนแปลงของค่าข้อมูล

10. กด Esc ออกมาที่เมนูหลัก แล้วเข้าไปที่หน้าจอ " Trend Display " กด Tab ไปที่ " TAG " คีย์คำว่า " TREND1 " กด Enter แล้วคีย์คำว่า " TWO TANKS " กด Enter หลังจากนั้นเลือก " SCAN TIME " เป็น " 1 SEC " โดยกด PageUp หรือ PageDown แล้วคีย์คำว่า " UNIT1 " กด Enter ต่อจากนั้น คีย์คำว่า " TANK2 " กด Enter จากนั้น ตั้งตัวแปรเป็น " MV " โดยใช้คีย์ PageUp หรือ PageDown ทำในลักษณะเดียวกันของ Node ต่อไปให้มีค่าตามรูป แล้วสังเกตการเปลี่ยนแปลงของค่าข้อมูล

(หมายเหตุ Trend จะแก้ไขค่าได้ใน ขณะที่ " TREND OFF " เท่านั้น)

11. กด Esc ออกมาที่เมนูหลัก แล้วเข้าไปที่หน้าจอ " Alarm List " กด Enter สังเกตดูค่าการเกิด Alarm ของบล็อกต่างๆ ว่ามีเป็นชนิดไหนและเกิดเมื่อไร

12. กด Esc ออกมาที่เมนูหลัก แล้วเข้าไปที่หน้าจอ " Group Display " เลื่อนเมาส์ไปคลิก หรือกด Tab เลือกที่คำว่า " TAG " ใส่ " GROUP1 " กด Enter แล้วคีย์คำว่า " TWO TANKS " ที่ " DESCRIPTOR " หลังจากนั้นคีย์ชื่อของ Node และ Tag ให้มีลักษณะดังรูป แล้วสังเกตการเปลี่ยนแปลงของค่าข้อมูล



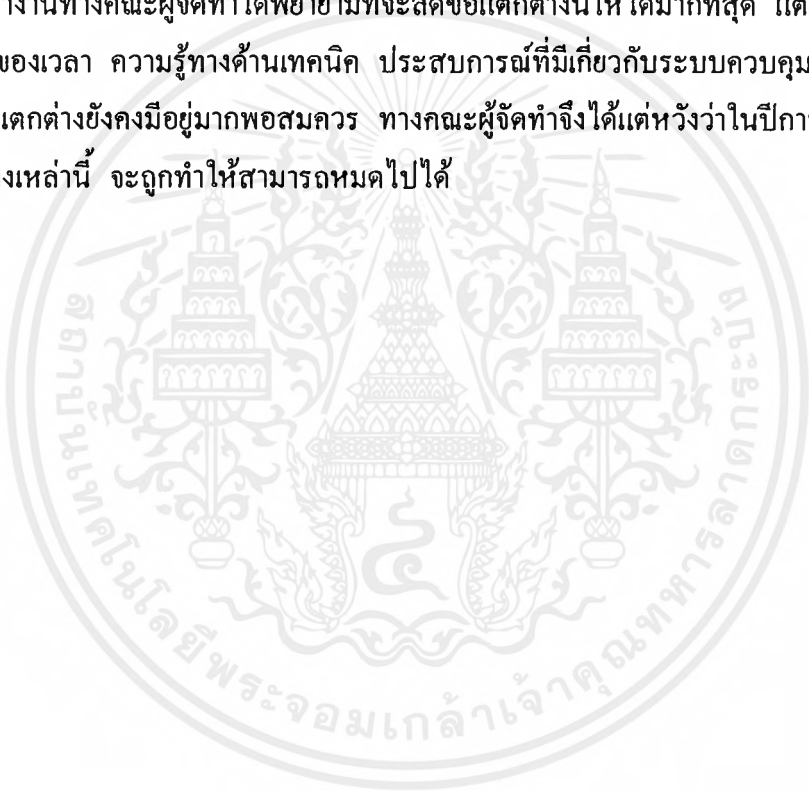
รูปที่ 5.4 หน้าจอ Group Display หลังจากทำการ Run Mode

ขั้นที่ 5 การบันทึกและการออกจากโปรแกรม

1. เลือกไปที่ " System Configuration " กด Enter เลือกเข้าไปที่ " System Memory " คีย์คำว่า " GP1.SYS "
2. เลือกไปที่ " System Configuration " กด Enter เลือกเข้าไปที่ " Group Configuration " คีย์คำว่า " GP1.SYS " เลือกไปที่ " WRITE " กด Enter
3. ทำเช่นเดียวกับข้อ 2 ในส่วนที่เป็น Trend Configuration , Alarm List
4. เลือกไปที่ "STANDBY MODE" กด Enter แล้วเลือกไปที่ "NETWORK SYSTEM" แล้วให้สังเกตดูว่ามี Node ใดอยู่ในระบบบ้าง เป็นชนิดใด หลังจากนั้น กด Esc เลือกไปที่ " Exit " กด Enter เพื่อจบการทำงานของ Program DCS

บทที่ 6 สรุปและวิจารณ์

จากการทำงานในการปีการศึกษา 2540 ที่ผ่านมา จะเห็นได้ว่าทางคณะผู้จัดทำได้พยายามที่จะจำลองและเลียนแบบระบบการควบคุมแบบกระจายส่วน (Distributed Control System) ให้มีความสามารถในการควบคุมเหมือนระบบที่ใช้งานจริงในวงการอุตสาหกรรม แต่เนื่องจากการทำงานชิ้นนี้ยังอยู่ในขั้นของโครงการในการเรียน (Thesis) ทำให้งานที่ออกมายังมีความแตกต่างในทางที่ด้อยกว่าระบบที่ใช้จริงอยู่มาก ไม่ว่าจะเป็นในด้านฟังก์ชันของการทำงานที่มีขึ้น เพื่อสนับสนุนเรื่องของความสะดวกในการทำงาน ความง่ายในการใช้งาน ความปลอดภัยในการใช้งาน หรือจะเป็นในด้านของรูปแบบความเฉพาะตัวของอุปกรณ์ ดังที่ได้กล่าวมาแล้วในบทที่ 3 ซึ่งตลอดระยะเวลาการทำงานทางคณะผู้จัดทำได้พยายามที่จะลดข้อแตกต่างนี้ให้ได้มากที่สุด แต่เนื่องจากปัญหาทางด้านของเวลา ความรู้ทางด้านเทคนิค ประสบการณ์ที่มีเกี่ยวกับระบบควบคุมแบบ DCS นั้นทำให้ความแตกต่างยังคงมีอยู่มากพอสมควร ทางคณะผู้จัดทำจึงได้แต่หวังว่าในปีการศึกษาต่อ ๆ ไปความแตกต่างเหล่านี้ จะถูกทำให้สามารถหมดไปได้



หนังสืออ้างอิง

[1] William A. Shay , “Understanding data communication and networks” PWS Publishing Company , a division of International Thomson Publishing Inc. ,1995 Printed in USA.

[2] David P. Ewing , “Client/Server LAN Programming” Que Corporation ,1994 Printed in USA.

