การควบคุมแบบดิจิตอล

DIGITAL CONTROL

โดย

นายกิตติพงษ์ แซ่ตั้ง

นายธีรเดช โชติก

วัน เดือน ปี 19 ส.ค. 2539
เลขทะเบียน 034928
เลขเรียกหนังสือ T 357028 ก6

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิศวกรรมระบบควบคุม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2537

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ปริญญานิพนธ์ ปีการศึกษา 2537

ภาควิชา วิศวกรรมระบบควบคุม

คณะ วิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหาร ลาดกระบัง

เรื่อง การควบคุมแบบดิจิตอล
DIGITAL CONTROL

ผู้จัดทำ นายกิตติพงษ์ แซ่ตั้ง รหัสประจำตัว 34101021
นายธีรเดช โชติก รหัสประจำตัว 34103153

รศ.สุเธียร เกียรติสุนทร
(อาจารย์ที่ปรึกษา)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การควบคุมแบบดิจิตอล
DIGITAL CONTROL

โดย นายกิตติพงษ์ แซ่ตั้ง รหัสประจำตัว 34101021
นายธีรเดช โชติก รหัสประจำตัว 34103153

อาจารย์ที่ปรึกษา รศ.สุเธียร เกียรติสุนทร

บทคัดย่อ

ปริญญานิพนธ์นี้มีวัตถุประสงค์ที่จะศึกษาทฤษฎี และการประยุกต์ใช้งานตัวควบคุมแบบดิจิตอล ในการควบคุมกระบวนการทางอุตสาหกรรม ในการทดลองจะใช้เครื่องไมโครคอมพิวเตอร์ IBM PC เป็นตัวควบคุมดิจิตอล โดยเชื่อมต่อกับกระบวนการโดยผ่านตัวแปลงสัญญาณอนาล็อกเป็นดิจิตอล และตัวแปลงสัญญาณดิจิตอลเป็นอนาล็อก เพื่อทำการควบคุมกระบวนการของถังผสมถ่ายเทความร้อน โดยอาศัยทฤษฎีการควบคุมแบบดิจิตอลและการควบคุมแบบป้อนล่วงหน้า (FEEDFORWARD CONTROL) เพื่อทำให้ระดับน้ำและอุณหภูมิของน้ำสามารถเข้าสู่เป้าหมายการควบคุมได้

ABSTRACT

The objective of this thesis is study in digital control strategy and it's application for the controling in industrial process. The microcomputer IBM PC is used in this experiment as digital controller by connecting with analog to digital converter and digital to analog converter for controling the process of heat exchanger tank . With the method of digital control and feedforward control , the setpoint of water level and water temperature can be reached.

กิตติกรรมประกาศ

ขอขอบพระคุณอาจารย์สุเธียร เกียรติสุนทร ให้คำแนะนำ อบรม สั่งสอนศิษย์ให้เป็นคนดี มีวิชาความรู้

ขอบพระคุณอาจารย์คงศักดิ์ อนันต์หิรัญรัตน์ ที่คอยติดตามดูแล

ขอบคุณพี่ติ๊ก (ธิดาพร) คอยให้คำปรึกษา และช่วยเหลือทุกอย่าง

ขอบคุณน้ำที่ให้เราควบคุมระดับและอุณหภูมิ และหล่อเลี้ยงชีวิตเราให้ดำรงอยู่ได้ และคอยให้กำลังใจมาตลอด

ขอบคุณเพื่อน ๆ และ น้องๆทุกคนช่วยคลายเหงา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทคัดย่อ(ABSTRACT)	I
กิตติกรรมประกาศ	II
สารบัญ	III
สารบัญภาพ	IV
บทที่ 1 บทนำ	1
บทที่ 2 ทฤษฎีการควบคุมแบบป้อนล่วงหน้า	2
• การออกแบบระบบควบคุมแบบป้อนล่วงหน้า	2
• เครื่องควบคุมแบบป้อนล่วงหน้า	3
• ระบบควบคุมแบบรวมการควบคุมแบบป้อนล่วงหน้าและป้อนกลับ	4
บทที่ 3 การเตรียมการ	12
1. การใช้โปรแกรม DCS	12
2. กระบวนการ	33
3. ส่วนเชื่อมต่อ(Interfaoe Board)	34
บทที่ 4 การดำเนินงาน	35
1. การควบคุมระดับน้ำ	35
2. การควบคุมอุณหภูมิของน้ำ	40
3. การควบคุมระดับและอุณหภูมิของน้ำ 1	45
4. การควบคุมระดับและอุณหภูมิของน้ำ 2	53
บทที่ 5 สรุปและวิจารณ์	63
เอกสารอ้างอิง	65
ภาคผนวก	66

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ

		หน้า
รูปที่ 2.1	ระบบควบคุมแบบป้อนล่วงหน้า	2
รูปที่ 2.2	ระบบควบคุมแบบป้อนล่วงหน้า	3
รูปที่ 2.3	ผลการควบคุมของระบบควบคุมแบบป้อนล่วงหน้า	5
รูปที่ 2.4	ผังสัญญาณของระบบควบคุมแบบป้อนล่วงหน้า	6
รูปที่ 2.5	ระบบควบคุมแบบรวมการควบคุมแบบป้อนล่วงหน้าและป้อนกลับ	10
รูปที่ 3.1	หน้าจอของเมนู	12
รูปที่ 3.2	การแสดงคำสั่งบนเมนู	14
รูปที่ 3.3	ตอนโทรลบล็อคไดอะแกรม	15
รูปที่ 3.4	Control Block Faceplate	17
รูปที่ 3.5	Edit System Memory	18
รูปที่ 3.6	Graphic Display	19
รูปที่ 3.7	หน้าจอของ Graphic Display Generator	20
รูปที่ 3.8	แสดงการกำหนดภาพจำลองใน EDIT ของ Graphic Display	21
รูปที่ 3.9	Group Display	22
รูปที่ 3.10	Input Block Configuration	22
รูปที่ 3.11	Output Block Configuration	23
รูปที่ 3.12	System Configuration	24
รูปที่ 3.13	System Memory Configuration	25
รูปที่ 3.14	Graphic Configuration ใน System Configuration	25
รูปที่ 3.15	Group Configuration ใน System Configuration	26
รูปที่ 3.16	Trend Configuration ใน System Configuration	26
รูปที่ 3.17	Trending Display	27
รูปที่ 3.18	หน้าจอของโปรแกรม Installation	29
รูปที่ 3.19	การกำหนดค่าเอาต์พุต	30
รูปที่ 3.20	การกำหนดค่าอินพุต	31
รูปที่ 3.21	Plant - 1301	33
รูปที่ 4.1	การควบคุมระดับน้ำ	35
รูปที่ 4.2	บล็อคไดอะแกรมแสดงการควบคุมน้ำ	35
รูปที่ 4.3	แสดงบล็อคการทำงานของโปรแกรม DCS ในการควบคุมอุณหภูมิน้ำ	38

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.4	รูปแสดงผลการทดลองการควบคุมระดับน้ำ	39
รูปที่ 4.5	การควบคุมระดับน้ำ	40
รูปที่ 4.6	บล็อคไดอะแกรมแสดงการควบคุมอุณหภูมิของน้ำ	40
รูปที่ 4.7	แสดงบล็อคการทำงานของโปรแกรม DCS ในการควบคุมอุณหภูมิน้ำ	43
รูปที่ 4.8	รูปแสดงผลการทดลองการควบคุมอุณหภูมิน้ำ	44
รูปที่ 4.9	การควบคุมระดับและอุณหภูมิน้ำ 1	45
รูปที่ 4.10	บล็อคไดอะแกรมแสดงการควบคุมระดับน้ำและอุณหภูมิน้ำ 1	45
รูปที่ 4.11	แสดงบล็อคการทำงานของโปรแกรม DCS ในการควบคุมอุณหภูมิและระดับน้ำ 1	51
รูปที่ 4.12	รูปแสดงผลการทดลองการควบคุมระดับและอุณหภูมิของน้ำ 1	52
รูปที่ 4.13	การควบคุมระดับและอุณหภูมิของน้ำ 2	53
รูปที่ 4.14	บล็อคไดอะแกรมแสดงการควบคุมระดับน้ำและอุณหภูมิน้ำ 2	53
รูปที่ 4.15	แสดงบล็อคการทำงานของโปรแกรม DCS ในการควบคุมอุณหภูมิและระดับน้ำ 2	61
รูปที่ 4.16	รูปแสดงผลการทดลองการควบคุมระดับและอุณหภูมิของน้ำ 2	62

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

ในปัจจุบันกระบวนการอุตสาหกรรมมีความยุ่งยากซับซ้อนมากขึ้น โดยเฉพาะในอุตสาหกรรมหนัก เช่น โรงกลั่นน้ำมัน,โรงปฏิกรณ์ปรมาณู เป็นต้น การควบคุมกระบวนการโดยใช้วงจรฮาร์ดแวร์มีความยุ่งยากและเสียค่าใช้จ่ายมาก ในการพัฒนาและเปลี่ยนแปลงไปตามสภาวะการณ์ ซึ่งประกอบกับในปัจจุบัน เทคโนโลยีทางด้านดอมพิวเตอร์ เจริญรุดหน้าทำให้เครื่องคอมพิวเตอร์มีราคาถูกลงมาก ในกระบวนการอุตสาหกรรมต่าง ๆ จึงนิยมนำคอมพิวเตอร์เข้ามาใช้ในการควบคุมกระบวนการต่าง ๆ อย่างแพร่หลาย เพราะทำให้สะดวกมากกว่า และง่ายต่อการพัฒนาและเปลี่ยนแปลงเพราะเราจะเปลี่ยนแปลงที่ซอฟต์แวร์ เท่านั้น

เนื่องจากคอมพิวเตอร์มีความสำคัญในการควบคุมทางอุตสาหกรรมดังกล่าว เราจึงได้นำโปรแกรมที่ได้พัฒนาขึ้นโดย รศ. สุเธียร เกียรติสุนทร มาใช้เพื่อทดลองควบคุมกระบวนการด้วยการควบคุมแบบดิจิตอลในแบบต่าง ๆ ซึ่งเราได้นำมาทดลอง 2 แบบ คือ การควบคุมแบบป้อนล่วงหน้า (FEEDFORWARD CONTROL) และ การควบคุมแบบหลายตัวแปรเข้า และหลายตัวแปรออก (DECOUPLING CONTROL) ซึ่งเป็นวิธีที่ใช้กันมากในกระบวนการอุตสาหกรรมในปัจจุบันเพื่อเป็นประโยชน์ทางการศึกษา การควบคุมกระบวนการด้วยวิธีควบคุมแบบดิจิตอลต่อไป

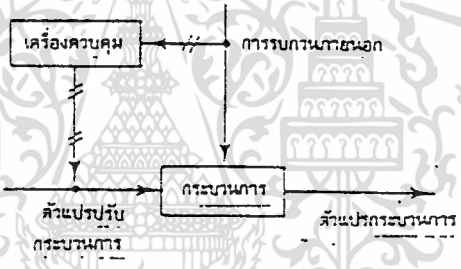
วัตถุประสงค์การดำเนินงาน เพื่อศึกษาทฤษฎี และประยุกต์ใช้งานตัวควบคุมแบบดิจิตอล ในการควบคุมกระบวนการทางอุตสาหกรรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ทฤษฎีการควบคุมแบบป้อนล่วงหน้า
(FEEDFORWARD CONTROL)

ระบบควบคุมแบบป้อนล่วงหน้าทำมาเพื่อแก้ไขข้อเสียของระบบควบคุมแบบป้อนกลับ โดยระบบควบคุมแบบป้อนล่วงหน้าจะตรวจสภาพแวดล้อม ที่มีสาเหตุทำให้เกิดความคลาดเคลื่อนในการควบคุมขึ้นโดยตรง และสัญญาณควบคุมจะเกิดขึ้นทันที เมื่อพบสาเหตุที่ทำให้มีสภาพกระบวนการเปลี่ยนแปลงเกิดขึ้น โดยสัญญาณควบคุมจะให้ผลการควบคุมหักล้างกับสาเหตุที่ทำให้สภาพกระบวนการเปลี่ยนแปลง การควบคุมแบบป้อนล่วงหน้าจึงสามารถรักษาสภาพของกระบวนการให้ตรงตามเป้าหมายการควบคุมตลอดเวลา



รูป 2.1 ระบบควบคุมแบบป้อนล่วงหน้า

<u>การออกแบบระบบควบคุมแบบป้อนล่วงหน้า</u>

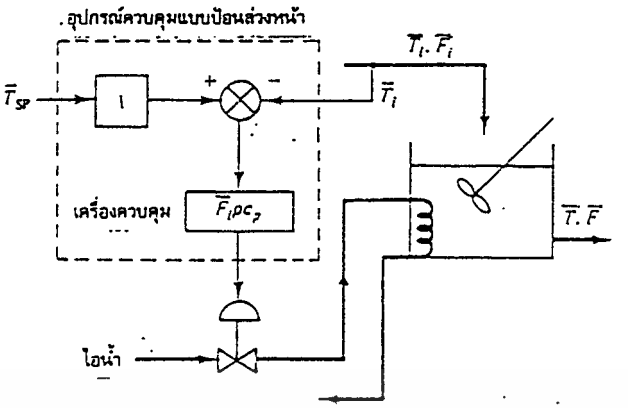
การออกแบบระบบควบคุมแบบป้อนล่วงหน้าสำหรับควบคุมถังผสมถ่ายเทความร้อนโดยพิจารณาแบบจำลองทางเณิตศาสตร์ของถังผสมถ่าเทความร้อน คือ

$$A \frac{dh}{dt} = Fi - F \qquad (2.1)$$

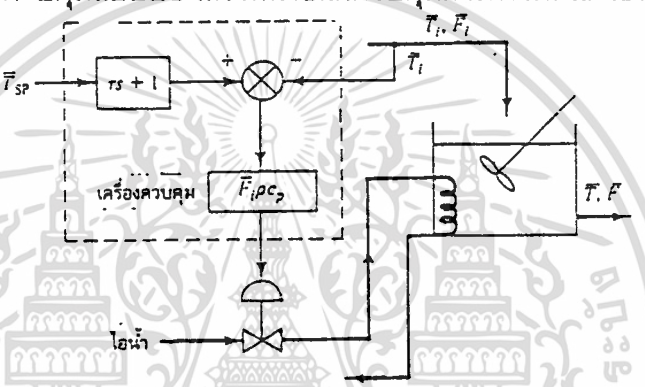
$$A\,h \frac{dT}{dt} = F_i(T_i - T) + \frac{Q}{\rho \cdot c_p} \qquad (2.2)$$

ถ้าอัตราการไหลเข้า F_i คงที่ และ $F_i = F$ จะทำให้ dh/dt=0 ถังผสมถ่ายเทความร้อนจะเหลือเพียงแบบจำลองทางคณิตศาสตร์แสดงการถ่ายเทความร้อนตามสมการ (2.2) อุณหภูมิของของเหลวที่ไหลเข้าสู่ถัง T_i เป็นการรบกวนภายนอก และปริมาณความร้อน Q เป็นสัญญาณปรับกระบวนการของถังผสมถ่ายเทความร้อนถ้าการควบคุมต้องการให้อุณภูมิของเหลวภายในถัง T มีค่าคงที่ตามเป้าหมายการควบคุม T_{sp}

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ก)ระบบควบคุมแบบป้อนล่วงหน้าชนิดควบคุมการตอบสนองเชิงสถิต



(ข)ระบบควบคุมแบบป้อนล่วงหน้าชนิดควบคุมการตอบสนองเชิงพลวัต

รูปที่ 2.2 ระบบควบคุมแบบป้อนล่วงหน้า

1. เครื่องควบคุมแบบป้อนบล่วงหน้าชนิดควบคุมการตอบสนองเชิงสถิต (steady state feedforward controller) สามารถทำได้โดยพิจารณาสภาพการตอบสนองสุดท้ายภายหลังกระบวนการเข้าสู่สภาพสมดุลทางความร้อน และอุณหภูมิของถังผสมถ่ายเทความร้อนมีค่าคงที่ หรือ dT/dt =0

$$o = Fi\ (Fi - T) + \frac{Q}{\rho c_p} \tag{2.3}$$

$$T = T_i + \frac{Q}{F_i . \rho . c_p} \tag{2.4}$$

ถ้าระบบควบคุมแบบป้อนล่วงหน้าสามารถทำให้อุณหภูมิของเหลวภายในถัง T มีค่าคงที่เท่ากับเป้าหมายการควบคุมT_{sp}จะได้แบบจำลองทางคณิตศาสตร์ของเครื่องควบคุมแบบป้อนล่วงหน้าตามสมการ (2.5)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$Q = F_i\ \rho c_p\ (T_{sp} - T_i) \tag{2.5}$$

รูปที่ 2.2(ก) แสดงลักษณะของระบบควบคุมแบบป้อนล่วงหน้าชนิดควบคุมการตอบสนองเชิงสถิต โดยปริมาณความร้อนจากไอน้ำ Q จะเปลี่ยนแปลงตามอุณหภูมิเป้าหมาย T_{sp}และอุณหภูมิของเหลวที่ไหลเข้าสู่ถัง T_i ตลอดเวลา เพื่อรักษาสภาพสมดุลสุดท้ายของถังผสมถ่ายเทความร้อนให้มีอุณหภูมิตรงกับอุณหภูมิเป้าหมายโดยไม่สนใจสภาพการตอบสนองชั่วขณะของกระบวนการในขณะเวลาเริ่มต้นของการเปลี่ยนแปลง

2. เครื่องควบคุมแบบป้อนล่วงหน้าชนิดควบคุมการตอบสนองเชิงพลวัต (dynamic feedforward controller) เพื่อปรับปรุงผลการตอบสนองรวมของระบบควบคุมแบบป้อนล่วงหน้าให้ดีขึ้น โดยพิจารณาแบบจำลองทางคณิตศาสตร์แสดงการแลกเปลี่ยนความร้อนของถังผสมถ่ายเทความร้อนตามสมการ (2.2)

$$A\,h\,\frac{dT}{dt} = F_i(T_i - T) + \frac{Q}{\rho \cdot c_p}$$

แทนค่า V=Ah เพื่อแสดงปริมาตรของเหลวภายในถัง

$$\frac{V}{F_i}\frac{dT}{dt} + T = T_i + \frac{Q}{F_i \rho\, c_p} \qquad (2.6)$$

แสดงตัวแปรต่างๆของสมการ (2.6)ในรูปของตัวแปรเบี่ยงเบน

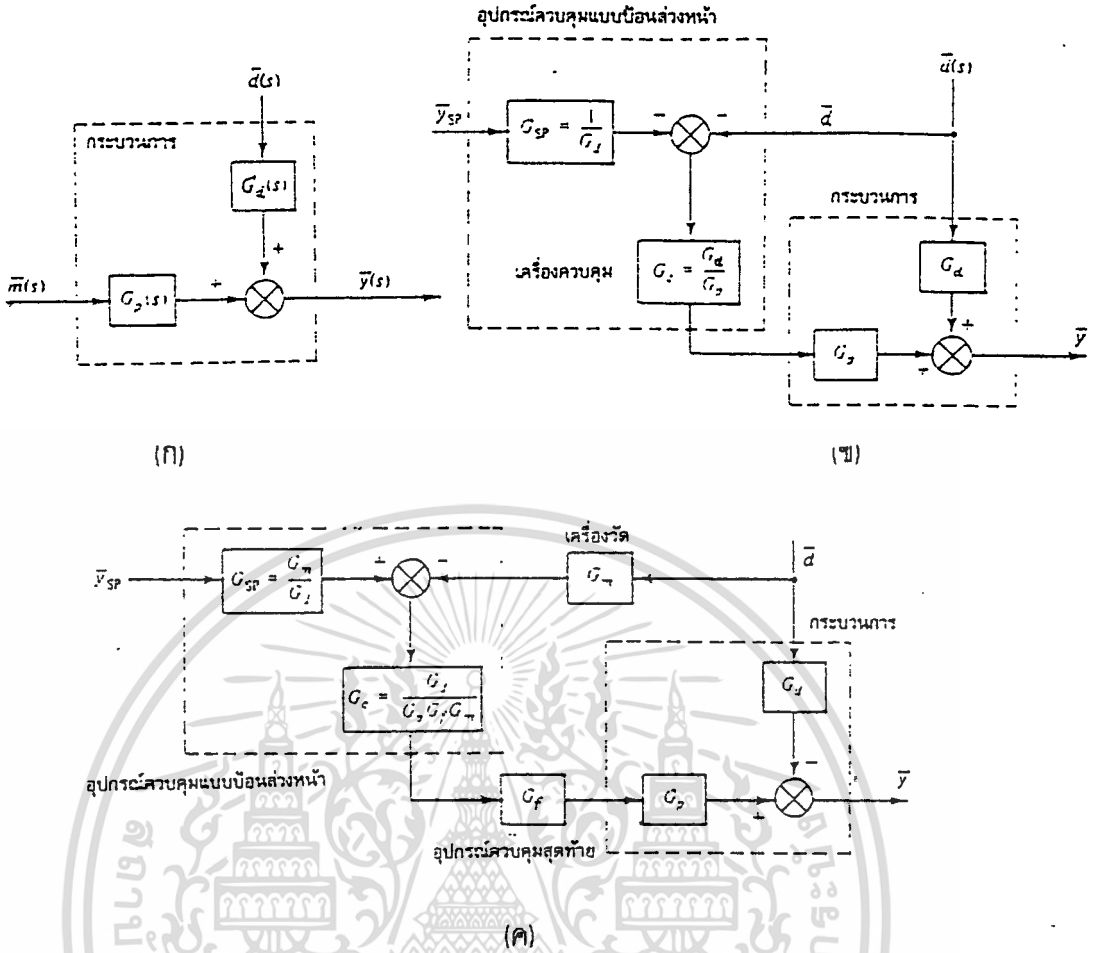
$$\frac{V}{F_i}\frac{dT'}{dt} + T' = T'_i + \frac{Q}{F_i \rho\, c_p} \qquad (2.7)$$

แปลงลาปลาซทั้งสองด้านของสมการ(2.7)

$$T'(s) = \frac{T'_i(s)}{\tau s + 1} + \frac{1}{F_i \rho c_p}\frac{1}{\tau s + 1} \qquad (2.8)$$

โดยτ =V/F_i ค่าคงที่การหน่วงเวลาของของเหลวภายในถังผสมถ่ายเทความร้อน ถ้าระบบควบคุมแบบป้อนล่วงหน้าสามารถทำให้อุณหภูมิของเหลวภายในถังT'มีค่าคงที่เท่ากับเป้าหมายการควบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.4 ผังสัญญาณของระบบควบคุมแบบป้อนล่วงหน้า (ก) กระบวนการ (ข) ระบบควบคุมแบบป้อนล่วงหน้า (ค) ระบบควบคุมแบบป้อนล่วงหน้าที่ประกอบด้วยเครื่องวัดและอุปกรณ์ควบคุมสุดท้าย

สัญญาณควบคุม m(s) ของเครื่องควบคุมแบบป้อนล่วงหน้าสามารถหาได้จากสมการ(2.11)

$$m(s) = \left[\frac{1}{Gd(s)} ysp(s) - d(s) \right] \frac{Gd(s)}{Gp(s)} \qquad (2.12)$$

$$m(s) = \left[Gsp(s) ysp(s) - d(s) \right] Gc(s) \qquad (2.13)$$

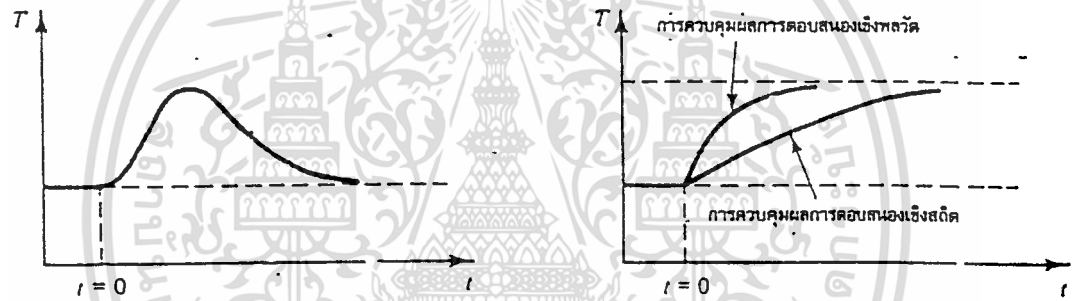
สมการ(2.13) แสดงแบบจำลองทางคณิตศาสตร์ของระบบควบคุมแบบป้อนล่วงหน้า ดังแสดงในรูปที่ 2.4 (ข) โดย $G_c(s)$ คือ ทรานส์เฟอร์ฟังก์ชั่นของหน่วยควบคุมแบบป้อนล่วงหน้าหลัก (main feedforward controller) และ $G_{sp}(s)$ คือ ทรานส์เฟอร์ฟังก์ชั่นของหน่วยควบคุมส่วนเป้าหมายการควบคุม (set point element)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คุม T'_{SP} ตลอดเวลาจะได้แบบจำลองทางคณิตศาสตร์ของเครื่องควบคุมแบบป้อนล่วงหน้าตามสมการ (2.9)

$$Q'(s) = Fi\rho cp[(\tau s + 1)T'_{sp}(s) - T'i(s)] \qquad (2.9)$$

รูปที่ 2.2 (ข) แสดงโครงสร้างของระบบควบคุมแบบป้อนล่วงหน้าชนิดควบคุมการตอบสนองเชิงพลวัต จะพบว่าเป้าหมายการควบคุม $T'_{SP}(s)$ ในสมการ (2.9) คูณกับพจน์ของทรานส์เฟอร์ฟังก์ชั่น ($\tau S+1$) ทำให้เครื่องควบคุมแบบป้อนล้วงหน้าชนิดควบคุมการตอบสนองเชิงสถิตในขณะที่มีการเปลี่ยนแปลงเป้าหมายการควบคุมเท่านั้น สำหรับการเปลี่ยนแปลงของการรบกวนจากภายนอกจะทำให้ได้ผลการควบคุมของระบบควบคุมแบบป้อนล่วงหน้าทั้งสองชนิดเปรียบเทียบกัน



(ก) ผลการควบคุมเมื่อการรบกวนภายนอกเปลี่ยนแปลง (ข) ผลการควบคุมเมื่อเป้าหมายการควบคุมเปลี่ยน แปลง

รูปที่ 2.3 ผลการควบคุมของระบบควบคุมแบบป้อนล่วงหน้า

การออกแบบระบบควบคุมแบบป้อนล่วงหน้าทั่วไปสามารถทำได้โดยพิจารณาทรานส์เฟอร์ฟังก์ชั่นของกระบวนการตามผังสัญญาณรูปที่ 2.4 (ก)

$$y(s) = G_p(s)m(s) + G_d(s)d(s) \qquad (2.10)$$

ถ้าระบบควบคุมแบบป้อนล่วงหน้าสามารถควบคุมตัวแปรกระบวนการ $y(s)$ ให้มีค่าเท่ากับเป้าหมายการควบคุม $y_{SP}(s)$

$$y_{sp}(s) = G_p(s)m(s) + G_d(s)d(s) \qquad (2.11)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$G_c(s) = \frac{G_d(s)}{G_p(s)} \tag{2.14}$$

$$G_{sp}(s) = \frac{1}{G_d(s)} \tag{2.15}$$

ระบบควบคุมแบบป้อนล่วงหน้า ดังแสดงในรูปที่ 2.4(ข) ตามสมการ (2.13) สมการ (2.14) และสมการ (2.15) มีคุณสมบัติต่างๆ ที่น่าสนใจดังนี้

1. ระบบควบคุมแบบป้อนล่วงหน้ามีส่วนประกอบต่างๆ ทั้งหมดเช่นเดียวกับระบบควบคุมแบบป้อนกลับแต่มีการจัดโครงสร้างของวงควบคุมต่างกัน

2. เครื่องควบคุมแบบล่วงหน้าตามสมการ (2.14) และสมการ (2.15) แสดงว่าระบบควบคุมแบบป้อนล่วงหน้าไม่สามารถใช้เครื่องควบคุมแบบป้อนกลับทั่วไป เช่น เครื่องควบคุมแบบพีพีไอ หรือ พีไอดี ควบคุมกระบวนการได้ โดยระบบควบคุมแบบป้อนล่วงหน้าจะต้องใช้อุปกรณ์ควบคุมพิเศษที่มีความสามารถในการคำนวณทางคณิตศาสตร์เป็นพิเศษเฉพาะต่างจากเครื่องควบคุมที่ใช้กับระบบควบคุมแบบป้อนกลับทั่วไป

3. การออกแบบระบบควบคุมแบบป้อนล่วงหน้าตามสมการ (2.14) และสมการ (2.15) จำเป็นต้องทราบแบบจำลองทางคณิตศาสตร์และทรานส์เฟอร์ฟังก์ชั่นที่ถูกต้องของกระบวนการ $G_p(s)$ และ $G_d(s)$

4. ระบบควบคุมแบบป้อนล่วงหน้าตามผังสัญญาณรูปที่ 2.4(ข) ไม่ได้นำทรานส์เฟอร์ฟังก์ชั่นของเครื่องวัดและอุปกรณ์ควบคุมสุดท้ายมาร่วมพิจารณา ทรานส์เฟอร์ฟังก์ชั่นของระบบควบคุมแบบป้อนล่วงหน้าทั่วไปสามารถหาได้โดยพิจารณาค่าตัวแปรกระบวนการ y(s) จากผังสัญญาณของระบบควบคุมแบบป้อนล่วงหน้า ดังแสดงในรูปที่ 2.4(ค) โดยนำทรานส์เฟอร์ฟังก์ชั่นของเครื่องวัดและอุปกรณ์ควบคุมสุดท้ายมาใช้ร่วมในการพิจารณาด้วย คือ

$$y(s) = G_p(s)\, G_f(s)\, G_c(s)\, G_{SP}(s) + [G_d(s) - G_p(s) G_f(s) G_c(s) G_m(s)] d(s) \tag{2.16}$$

ทรานส์เฟอร์ฟังก์ชั่นของ $G_p(s)$ และ $G_{sp}(s)$ สามารถหาได้โดยพิจารณาวัตถุประสงค์ในการออกแบบระบบควบคุมแบบป้อนล่วงหน้า คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. ลดผลการรบกวนจากสภาพแวดล้อมภายนอก ระบบควบคุมแบบป้อนล่วงหน้าจะต้องสามารถลดผลการรบกวนจากการเปลี่ยนแปลงของสภาพแวดล้อมภายนอก โดยสัมประสิทธิ์ของ d(s) ในสมการ (16) จะต้องมีค่าเท่ากับ 0

$$G_d(s) - G_p(s)G_f(s)G_c(s)G_m(s) = 0$$

ทรานส์เฟอร์ฟังก์ชั่นของหน่วยควบคุมแบบป้อนลวงหน้าหลัก คือ

$$G_c(s) = \frac{G_d(s)}{G_p(s)G_f(s)G_m(s)} \tag{2.17}$$

2. การควบคุมตัวแปรกระบวนการให้มีค่าเท่ากับเป้าหมายการควบคุม ระบบควบคุมแบบป้อนล่วงหน้าจะต้องรักษาสภาพของกระบวนการ y(s) ให้มีค่าเท่ากับเป้าหมายการควบคุมที่ต้องการ $y_{sp}(s)$ โดยสัมประสิทธิ์ของ $y_{sp}(s)$ ในสมการ (2.16) จะต้องมีค่าเท่ากับ 1

$$G_p(s)G_f(s)G_c(s)G_{sp}(s) = 1$$

$$G_p(s)G_f(s)\left(\frac{G_d(s)}{G_p(s)G_f(s)G_m(s)}\right)G_{sp}(s) = 1$$

ทรานส์เฟอร์ฟังก์ชั่นของหน่วยควบคุมส่วนเป้าหมายการควบคุม คือ

$$G_{sp}(s) = \frac{G_m(s)}{G_d(s)} \tag{2.18}$$

<u>เครื่องควบคุมแบบป้อนล่วงหน้า</u>

การออกแบบระบบควบคุมแบบป้อนล่วงหน้าพบว่าเครื่องควบคุมแบบป้อนล่วงหน้าไม่สามารถใช้เครื่องควบคุมแบบทั่วไปได้เช่นเดียวกับระบบควบคุมแบบป้อนกลับโดยเครื่องควบคุมจะต้องมีความสามารถในการคำนวณทางคณิตศาสตร์ตามสมการ (2.14) และสมการ (2.15) ปัจจุบันระบบควบคุมทั่วไปจึงนิยมให้เครื่องคอมพิวเตอร์ทำหน้าที่เป็นเครื่องควบคุมแบบป้อนล่วงหน้า สำหรับระบบควบคุมที่ไม่ได้ใช้เครื่องคอมพิวเตอร์แทนเครื่องควบคุมแบบป้อนล่วงหน้าการสร้างอุปกรณ์ควบคุมให้มีความสามารถในการคำนวณตามทรานส์เฟอร์ฟังก์ชั่นของเครื่องควบคุม

เอกสารนี้ ... ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
034728

แบบป้อนล่วงหน้าเป็นเรื่องยุ่งยากและต่องเสียค่าใช้จ่ายมาก การออกแบบระบบควบคุมจึงพยายามใช้อุปกรณ์อื่นทำหน้าที่เป็นเครื่องควบคุมแบบป้อนล่วงหน้าโดยพิจารณาทรานส์เฟอร์ฟังก์ชั่นของเครื่องควบคุมแบบป้อนล่วงหน้า $G_c(s)$ และ $G_{sp}(s)$ ตามสมการ (2.14) และสมการ (2.15) ซึ่งอยู่ในรูปทรานส์เฟอร์ฟังก์ชั่นของกระบวนการ คือ $G_p(s)$ และ $G_d(s)$ เมื่อพิจารณาทรานส์เฟอร์ฟังก์ชั่นของกระบวนการจากตัวอย่าง พบว่าทรานส์เฟอร์ฟังก์ชั่นของกระบวนการทั่วไปสามารถแบ่งออกเป็น 2ส่วน คือ อัตราขยายเชิงสถิตของกระบวนการ และทรานส์เฟอร์ฟังก์ชั่นแสดงการตอบสนองชั่วขณะของกระบวนการ คือ

$$G_p(s) = \frac{1}{F_i \rho c_p} \frac{1}{\tau s + 1}$$

แทนค่า อัตราขยายเชิงสถิตของกระบวนการ $K_p = 1/F_i\rho c_p$ และทรานส์เฟอร์ฟังก์ชั่นแสดงการตอบสนองชั่วขณะของกระบวนการ $G_p'(s) = 1/\tau S+1$ จะสามารถแสดงทรานส์เฟอร์ฟังก์ชั่นของกระบวนการใหม่

$$G_p(s) = K_p G_p'(s) \qquad (2.19)$$

ทำนองเดียวกัน สามารถแสดงทรานส์เฟอร์ฟังก์ชั่น $G_d(s)$ ในรูปใหม่ คือ

$$G_d(s) = K_d G_d'(s) \qquad (2.20)$$

เมื่อ $K_d = 1$ และ $G_d'(s) = 1/\tau S + 1$

1. เครื่องควบคุมแบบป้อนล่วงหน้าชนิดควบคุมการตอบสนองเชิงสถิตของกระบวนการในตัวอย่างที่ 2 แสดงว่า $G_p(s) = K_p$และ$G_d = K_d$ ทรานส์เฟอร์ฟังก์ชั่นของเครื่องควบคุมแบบป้อนล่วงหน้า คือ

$$G_c(s) = K_d / K_p \qquad (2.21)$$

และ

$$G_{sp}(s) = 1 / K_d \qquad (2.22)$$

ทรานส์เฟอร์ฟังก์ชั่นของเครื่องควบคุมแบบป้อนล่วงหน้าชนิดควบคุมการตอบสนองเชิงสถิตของกระบวนการเป็นเพียงค่าคงที่จึงสามารถใช้เครื่องควบคุมแบบพีทั่วไปทำหน้าที่เป็นเครื่องควบคุมแบบป้อนล่วงหน้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. เครื่องควบคุมแบบป้อนล่วงหน้าชนิดควบคุมการตอบสนองเชิงพลวัตของกระบวนการในตัวอย่างที่ 2 แสดงว่า $G_p(s)$ และ $G_d(s)$ มีคุณสมบัติเป็นทรานส์เฟอร์ฟังก์ชั่นของกระบวนการหน่วงเวลาอันดับหนึ่งและทรานส์เฟอร์ฟังก์ชั่นของเครื่องควบคุมแบบป้อนล่วงหน้า คือ

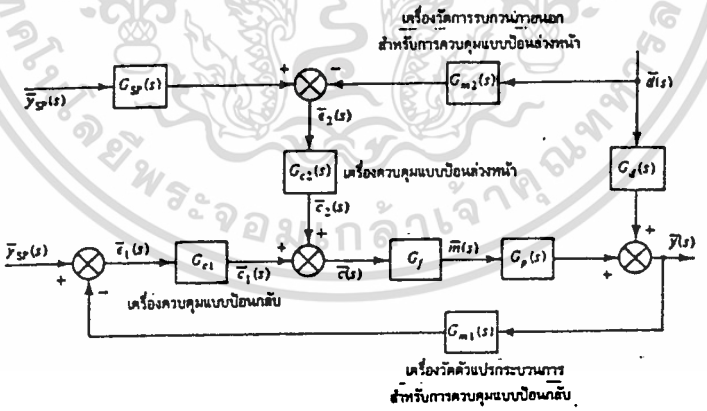
$$G_c(s) = \frac{G_d(s)}{G_p(s)} = \frac{1/(\alpha s+1)}{1/(\beta s+1)} = \frac{\beta s+1}{\alpha s+1} \qquad (2.23)$$

และ

$$G_{sp}(s) = \frac{1}{G_d(s)} = \frac{1}{1/(\alpha s+1)} = \alpha s+1 \qquad (2.24)$$

ทรานส์เฟอร์ฟังก์ชั่นของเครื่องควบคุมแบบป้อนล่วงหน้าชนิดควบคุมการตอบสนองเชิงพลวัตของกระบวนการสามารถใช้อุปกรณ์ชดเชยแบบมุมนำและอุปกรณ์ชดเชยแบบมุมตามทั่วไปทำหน้าที่เป็นเครื่องควบคุมแบบป้อนล่วงหน้า

ระบบควบคุมแบบรวมการควบคุมแบบป้อนล่วงหน้าและป้อนกลับ



รูปที่ 2.5 ระบบควบคุมแบบรวมการควบคุมแบบป้อนล่วงหน้าและป้อนกลับ

ระบบควบคุมแบบรวมการคบุคมแบบป้อนล่วงหน้าและป้อนกลับ (feedforward - feedback control) สามารถทำได้โดยพิจารณาค่าของตัวแปรกระบวนการ y(s) จากผังสัญญาณของระบบควบคุมแบบรวมการควบคุมแบบป้อนล่วงหน้าและป้อนกลับรูปที่ 2.5

$$y(s) = G_p(s)m(s) + G_d(s)d(s) \qquad (2.25)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สัญญาณควบคุมรวมของเครื่องควบคุมทั้งสอง คือ

$$m(s) = G_f(s)\, c(s)$$

$$= G_f(s)[c_1(s) + c_2(s)]$$

$$= G_f(s)G_{c1}(s)\mathcal{E}_1(s) + G_f(s)G_{c2}(s)\mathcal{E}_2(s)$$

$$m(s) = G_f(s)G_{c1}(s)[y_{SP}(s) - G_{m1}(s)y(s)] + G_f(s)G_{c2}(s)[G_{SP}(s)y_{SP}(s) - G_{m2}(s)d(s)] \qquad (2.26)$$

แทนค่า m(s) จากสมการ (2.26) ในสมการ (2.25)

$$y(s) = \frac{G_p(s)G_f(s)\left[G_{c1}(s) + G_{c2}(s)G_{sp}(s)\right]}{1 + G_p(s)G_f(s)G_{c1}(s)G_{m1}(s)} y_{sp}(s)$$

$$+ \frac{G_d(s) - G_p(s)G_f(s)G_{c2}(s)G_{m2}(s)}{1 + G_p(s)G_f(s)G_{c1}(s)G_{m1}(s)} d(s) \qquad (2.27)$$

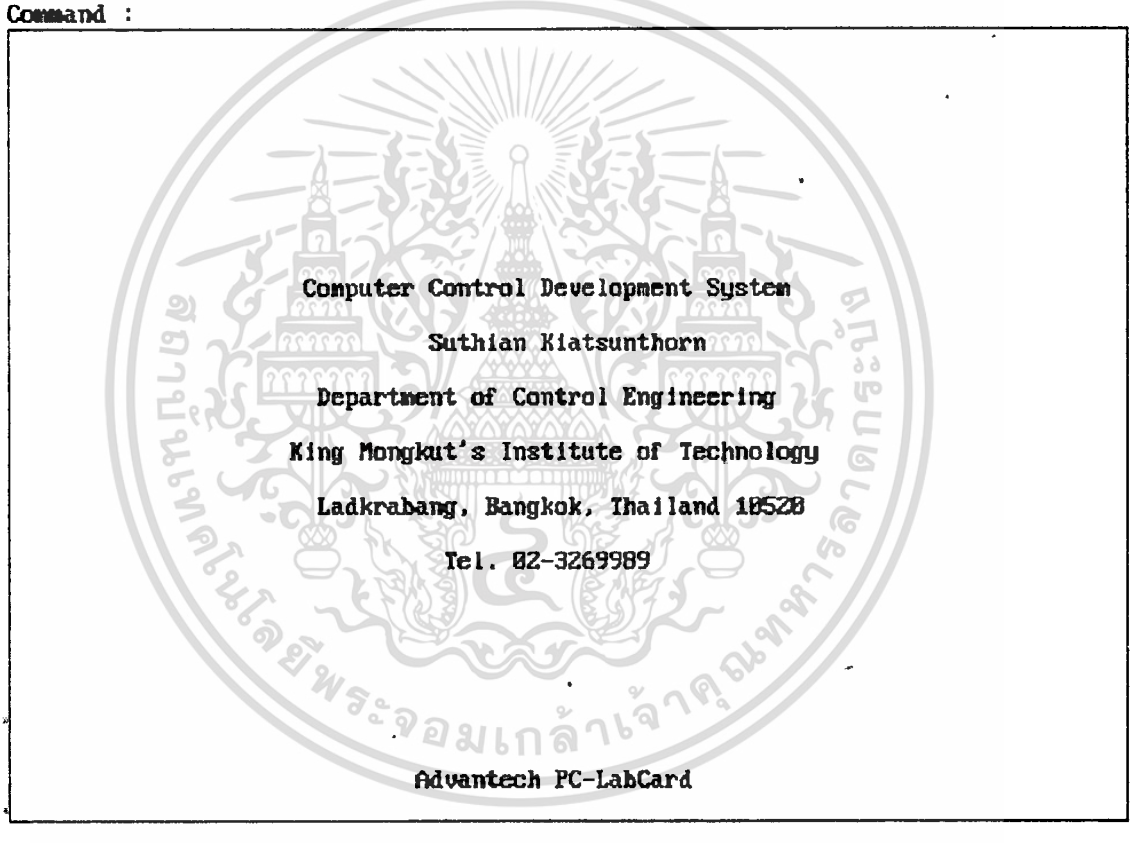
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
การเตรียมการ

1. การใช้โปรแกรม DCS

เมนู (MENU)

เมื่อเข้าโปรแกรม DCS จะปรากฏหน้าจอดังรูป ซึ่งหน้าจอนี้ จะเป็นหน้าจอที่สามารถเข้าไปที่เมนูได้ ดังนั้นก่อนที่จะเข้าไปที่หน้าจอของคำสั่งอื่นได้ จะต้องเข้ามาที่หน้าจอของเมนูนี้ก่อน จึงขอเรียกหน้านี้ว่า เมนู



รูปที่ 3.1 **หน้าจอของเมนู**

จากเมนูนี้จะสามารถเข้าไปที่คำสั่งต่างๆ ได้โดยกดคีย์[Tab] ซึ่งคำสั่งแรกที่ปรากฏขึ้น คือ คำสั่ง Alarm List และถ้ากด [Tab] อีกก็จะเป็นคำสั่งต่อไป เรียงไปตามพยัญชนะภาษาอังกฤษ ซึ่งมีคำสั่งต่างๆ เรียงไปดังนี้.

1. Alarm List
2. Control Block Diagram
3. Control Block Faceplate

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. Clear System Memory

5. Digital Block Diagram

6. Digital Block Faceplate

7. Edit System Memory

8. EXIT

9. Graphic Display

10.Graphic Display Generator

11.GRoup Display

12.Input Block Configuration

13.Initial System Memory

14.Output Block Configuration

15.Run Mode

16.Read System Memory

17.System Configuration

18.Sequential Control Monitor

19.Sequential Control Program

20.Standby Mode

21.Trending display

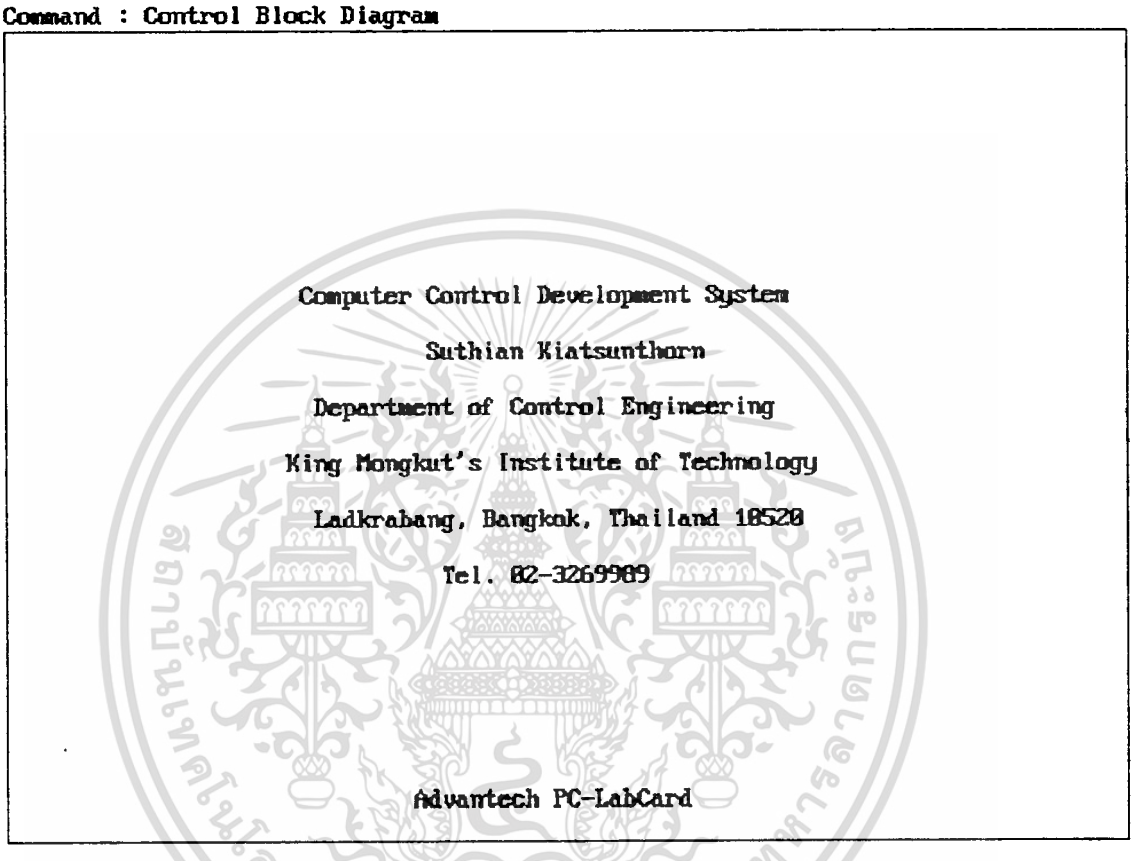
22.Write System Memary

นอกจากนี้ การเข้าไปที่คำสั่งดังกล่าวสามารถทำได้อีกวิธี ซึ่งจะเร็วกว่าวิธีแรก ซึ่งจะสังเกตเห็นว่าที่ เมนู แต่ละคำสั่งบนเมนูจะมีอักษรตัวใหญ่ปรากฏอยู่ เราสามารถกดคีย์ของตัวอักษรเหล่านั้น ก็จะปรากฏคำสั่งดังกล่าวที่ เมนู เช่น ถ้าเราต้องการเข้าไปที่ คอนโทรลบล็อคไดอะแกรม (Control Block Diagram) เราจะกดคีย์ [C][B][D] ก็จะปรากฏ "Control Block Diagram" ที่เมนู

เมื่อมีคำสั่งที่เราต้องการปรากฏบนเมนู เราจะสามารถเข้าไปในบล็อคดังกล่าวได้โดยกดคีย์ [Enter] เมื่อเข้าไปสู่ในบล็อคแต่ละบล็อคแล้ว การเข้าไปที่ตำแหน่งต่างๆในบล็อคแต่ละบล็อคนั้นจะใช้การกดปุ่ม [Tab] เป็นการเลื่อนเคอร์เซอร์ ซึ่งมีลักษณะเป็นแถบแดงไปตามตำแหน่ง แต่ถ้าต้องการลบแถบแดงออกไป ให้กดคีย์ [Esc] และในการออกจากบล็อคต่างๆ ก็

เอกสารนี้ [illegible] ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สามารถทำได้โดยการกดคีย์ [Esc] เช่นเดียวกัน ซึ่งเมื่อออกจากบล็อคของคำสั่งนั้นแล้ว จะปรากฏเป็นหน้าจอของเมนู ซึ่งสามารถเข้าสู่คำสั่งอื่นได้ต่อไป

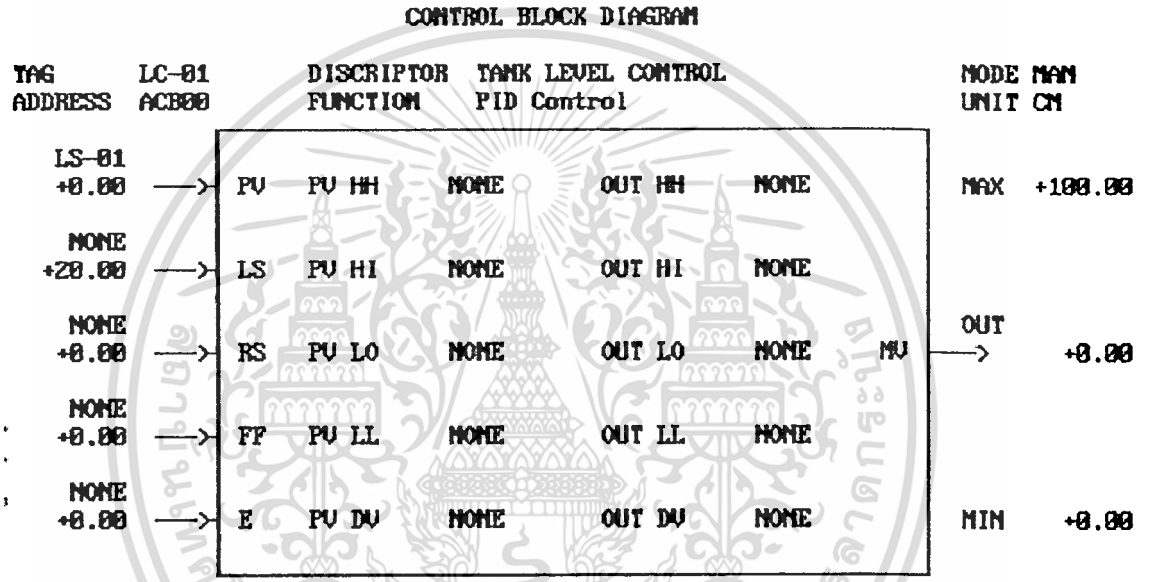


รูปที่ 3.2 **การแสดงคำสั่งบนเมนู**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำสั่ง (COMMAND)

1. Alarm list (เป็นคำสั่งที่ไม่ได้ใช้ในการทดลองนี้)
2. Control Block Diagram



รูปที่ 3.3 **คอนโทรลบล็อคไดอะแกรม**

ลักษณะที่แสดงของคอนโทรลบล็อคไดอะแกรม จะเป็นดังรูป ซึ่งจะขออธิบายความหมายของคำต่างๆ ที่ควรรู้ดังนี้

TAG คือชื่อของบล็อคไดอะแกรมที่แอดเดรสที่เฉพาะเจาะจงซึ่งได้ระบุไว้ เพื่อใช้กับการทำงานหนึ่ง เป็นชื่อที่นำไปใช้ลิงค์ต่อกับบล็อคการทำงานอื่น ซึ่งชื่อ TAG นี้ อาจจะไม่ใส่ก็ได้ ซึ่งการลิงค์จะสามารถลิงค์โดยระบุแอดเดรสโดยตรงได้ การกำหนด ชื่อ TAG นี้สามารถกำหนดเป็นตัวอักษรใดก็ได้ในคีย์บอร์ด

ADDRESS คือ แอดเดรสที่จองไว้สำหรับการทำงานหนึ่ง สามารถนำไปกำหนดในการลิงค์ต่อกับบล็อคการทำงานอื่น ซึ่งแอดเดรสของแต่ละบล็อคไดอะแกรม จะกำหนดโดยใช้ตัวอักษรต่างกัน ซึ่งสำหรับของคอนโทรลบล็อคไดอะแกรม จะใช้ AC800 ถึง AC899

DISCRIPTOR คือ รายละเอียดต่างๆ ที่ต้องการจะแสดงไว้ในบล็อคแต่ละบล็อค

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

FUNCTION คือ ฟังก์ชั่นของการควบคุมสำหรับ คอนโทรลบล็อคไดอะแกรม มี 20 ฟังก์ชั่น ซึ่งแต่ละฟังก์ชั่นจะมีสมการแสดงให้เห็นอยู่ใน Control Block Faceplate (ซึ่งเราสามารถเปลี่ยนกลับไปกลับมาระหว่างบล็อคของ Control Block Diagram กับบล็อคของ Control Block Faccplate ได้โดยตรงโดยไม่ต้องกลับไปที่เมนู โดยกดคีย์[X]) การเลือกฟังก์ชั่นสามารถทำได้โดยกดคีย์ [Pg Up] และ [Pg Dn] ซึ่งฟังก์ชั่นของการควบคุมทั้ง 20 ฟังก์ชั่น มีดังนี้

1. MAN Manual
2. ON/OFF Control
3. PID Control
4. LL Lead/Lag
5. DT Dead Time
6. TOT Totalizer
7. RB Ratio/Bias
8. SSW Signal Selector Switch
9. HSW High Selector Switch
10. LSW Low Selector Switch
11. PLI Piececaise Interpolaton
12. POCY Polynomial Function
13. Mass Storage Process
14. Stirred Tank Heater
15. Heat Thansfer Process
16. Chemical Reaction Rate
17. CSTR Mass Balance of A
18. CSTR Energy Balance
19. Miring Process Bal of A
20. Miring Process Energy Bal

MODE มี 3 โหมด คือ MAN = Manual

AUT = Auto

REM = Remote

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

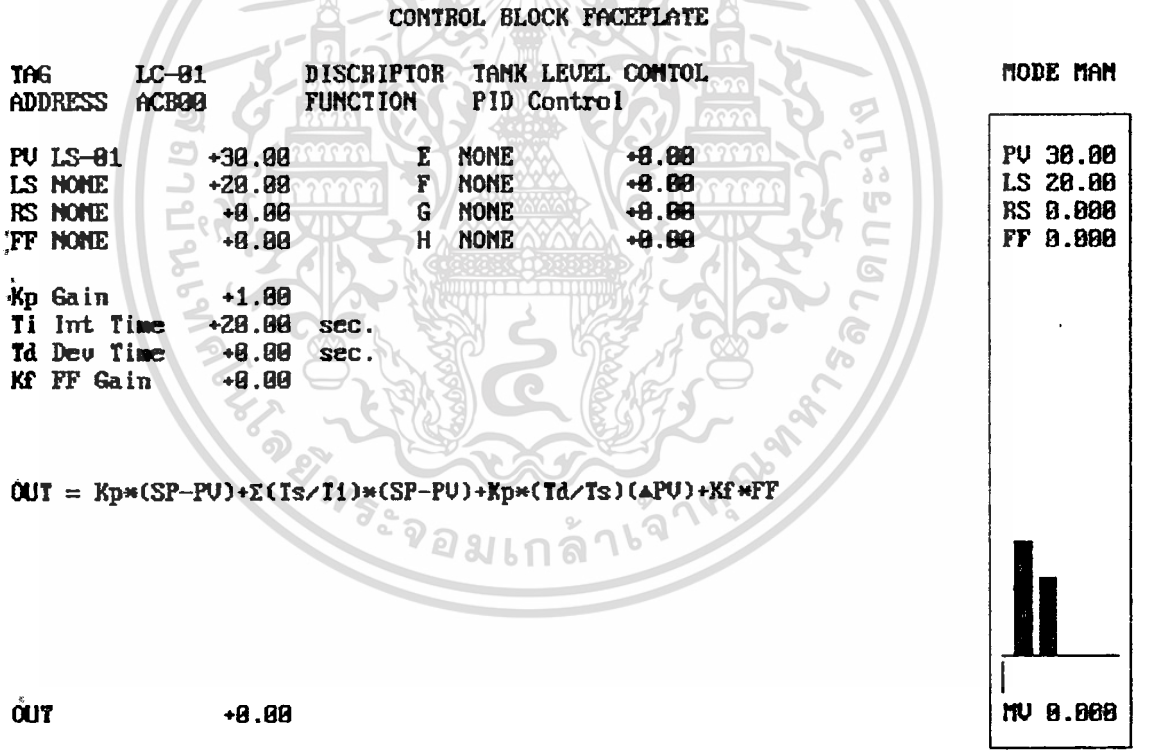
สามารถเลือกใช้โหมดทั้ง 3 โดยกดคีย์ [Pg Up] หรือ [Pg Dn] และกด [Enter]เมื่อโหมดที่แสดงอยู่นั้นเป็นโหมดที่ต้องการ

UNIT คือหน่วยที่ใช้ในบล็อคนั้น

<u>หมายเหตุ</u> สำหรับกรอบสี่เหลี่ยมด้านล่างจะมีลักษณะเปลี่ยนไปตาม FUNCTION

3. Control Block Faceplate

Control Block Faceplate จะเป็นส่วนที่ใช้แสดงค่าต่างๆของ คอนโทรลบล็อคไดอะแกรม ที่แอดเดรสเดียวกัน ซึ่งลักษณะของ Control Block Faceplate จะแตกต่างกันตามฟังก์ชั่นของการควบคุมเหมือนกับ ลักษณะของคอนโทรลบล็อคไดอะแกรม



รูปที่ 3.4 **Control Block Faceplate**

ซึ่งลักษณะที่แสดงในบล็อคนี้ก็จะมีอินพุต(PV), เซตพอยน์(LS,RS) พารามิเตอร์ต่างๆของตัวควบคุม และจะมีสมการแสดงฟังก์ชั่นการควบคุมในรูปแบบที่เลือก และมีค่าเอาต์พุตของบล็อคควบคุมแสดงที่ด้านล่างสุด ภายในFaceplateซึ่งมีลักษณะเป็นกรอบสี่เหลี่ยมทางด้านขวา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะมีกราฟแท่งแสดงค่าของตัวแปรในการควบคุม และแสดงค่าเป็นตัวเลขด้วยที่ด้านบนของ Faceplate

4. Clear System Memory คำสั่ง Clear System Memory จะเป็นการเคลียร์ข้อมูลที่เราเขียนไว้ในบล็อคต่างๆ จะทำให้บล็อคทุกบล็อคเป็นบล็อคเปล่า ที่ยังไม่มีการใส่ข้อมูลเข้าไป แต่คำสั่งนี้ไม่ได้กระทำกับ System memory หรือไฟล์ DCS.BLK แต่อย่างใด

5. Digital Block Diagram (เป็นคำสั่งที่ไม่ได้ใช้ในการทดลองนี้)

6. Digital Block Faceplate (เป็นคำสั่งที่ไม่ได้ใช้ในการทดลองนี้)

7. Edit System Memory

SYSTEM MEMORY

POSITION	DECIMAL	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 3.5 Edit System Memory

Edit System Memory จะบอกค่าปัจจุบันของตัวแปร ซึ่งเก็บอยู่ในหน่วยความจำของระบบ คือเก็บอยู่ในไฟล์ DCS.BLK ตัวแปรดังกล่าวคือ

I00 - I63

Q00 - Q63

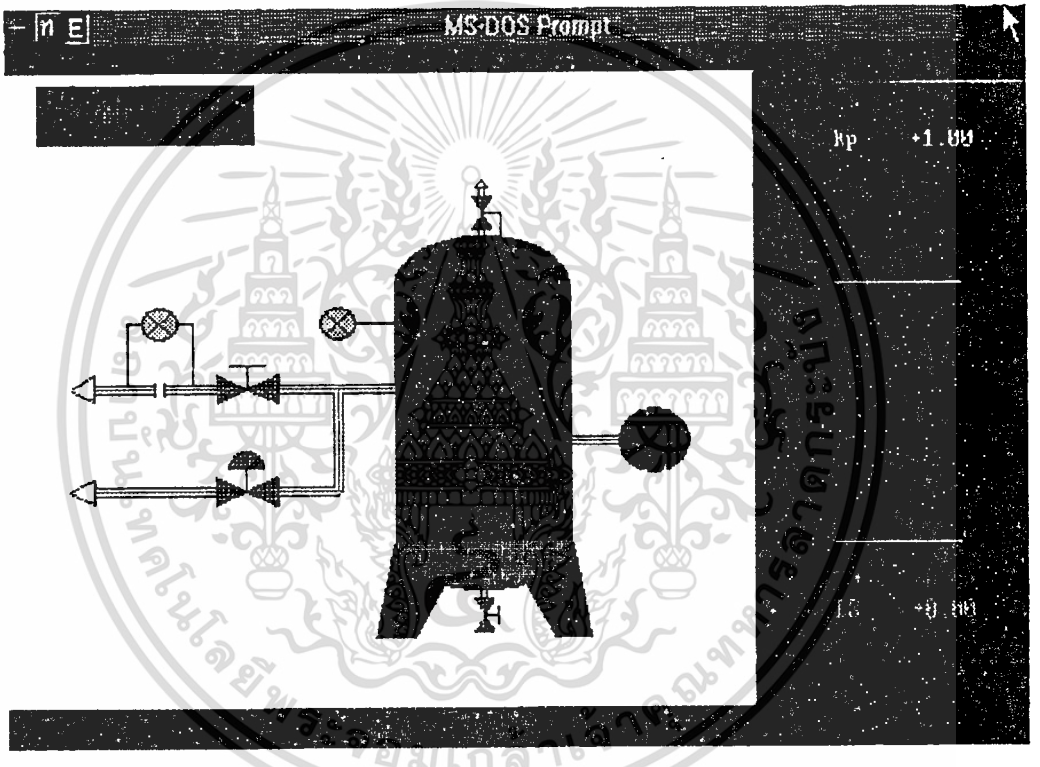
F00 - F63

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซึ่งจะแสดงออกมาเป็นเลขฐานสิบ (DECIMAL) , เลขฐานสิบหก (HEXADECIMAL)และเลขฐานสอง(BINARY) ซึ่งเราสามารถเลือกดูค่าตัวแปรต่างๆเหล่านี้ได้โดยกดคีย์ [Pg Up] หรือ [Pg Dn]

8. EXIT ใช้เพื่อออกจากโปรแกรม DCS. ซึ่งจะทำได้เมื่ออยู่ใน Standby Mode เท่านั้น ถ้าอยู่ใน Run Mode จะต้องเปลี่ยนเป็น Standby Mode ก่อน

9. Graphic Display



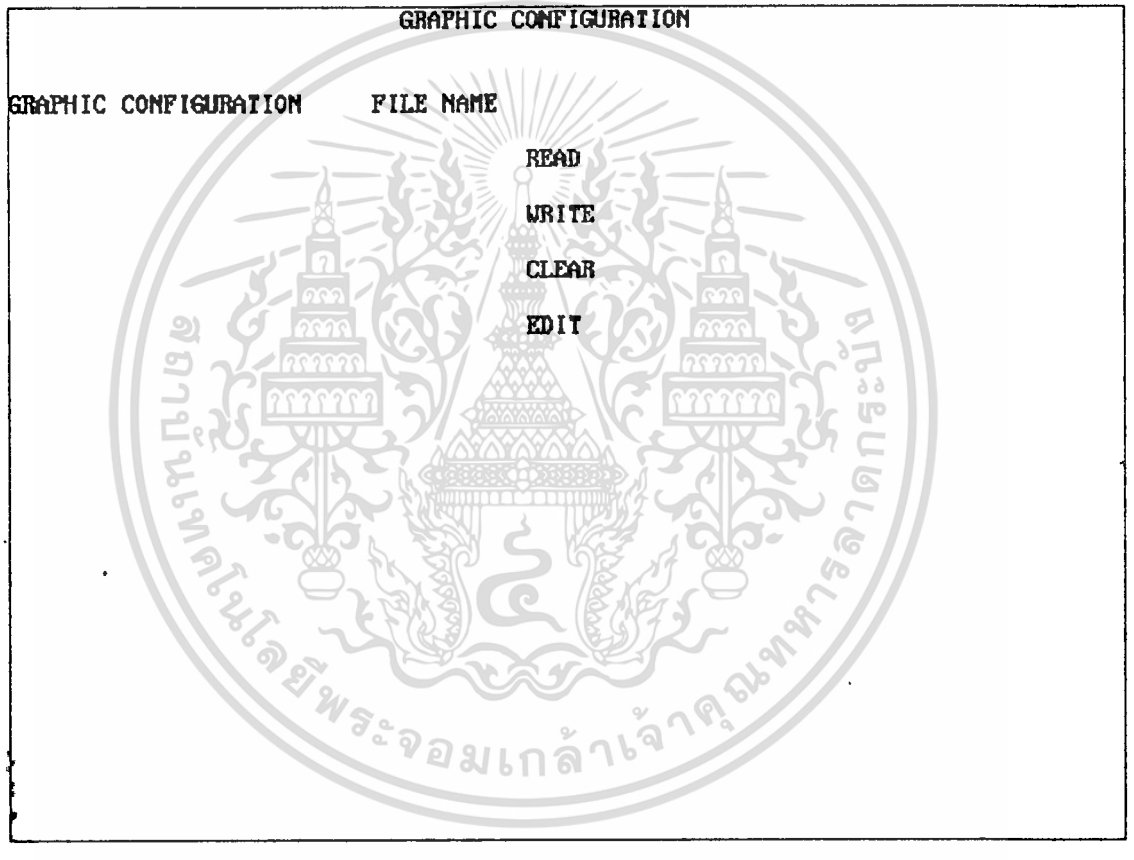
รูปที่ 3.6 **Graphic Display**

Graphic Display จะเป็นการแสดงภาพจำลองของกระบวนการ ซึ่งเป็นภาพจำลองที่เราสร้างขึ้นเพื่อแสดงให้เห็นลักษณะของกระบวนการ ตามค่าตัวแปรกระบวนการ ซึ่งเราสามารถเก็บภาพจำลองเพื่อแสดงใน Graphic Display ได้ไม่เกิน 10 ภาพ ภาพ Graphic Screen นี้ สามารถดูได้ทีละภาพโดยสามารถเปลี่ยนไปดูภาพอื่นโดยใช้การกด [Pg Up] หรือ [Pg Dn] ซึ่งก่อนที่จะนำภาพมาปรากฏใน Graphic Display ได้นี้ เราจะต้องเข้าไปกำหนดในคำสั่ง Graphic Display Generator ซึ่งจะได้กล่าวต่อไป ซึ่งครั้งแรกเลยเมื่อเราเข้ามาในคำสั่งของ Graphic Display นี้ ภาพแรกที่ปรากฏ จะเป็นภาพที่เรากำหนดไว้ใน Graphic Screen 1 ถ้ากด [Pg Up] ก็จะปรากฏ ภาพที่กำหนดไว้ใน Graphic Screen 2 หรือถ้ากด [Pg Dn] ก็จะปรากฏ ภาพที่กำหนดไว้ใน Graphic Screen 10 ซึ่งในกรณีตัวอย่างที่กล่าวมานี้ จะเกิดขึ้นได้ต้องกำหนด Graphic Screen ไว้ทั้ง 10 Graphic Screenใน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Graphic Display Generator ที่มุมล่างด้านซ้ายใน Graphic Display จะมีชื่อไฟล์ของภาพนั้นที่เรากำหนดไว้ใน Graphic Screen ดังกล่าว ดังรูปด้านบนชื่อไฟล์ของภาพที่แสดงอยู่คือ PROJECT4.OBJ ซึ่งภาพที่เรานำมาแสดงใน Graphic Display นี้ สามารถสร้างโดยใช้โปรแกรม GEDIT

10. Graphic Display Generator



รูปที่ 3.7 **หน้าจอของ Graphic Display Generator**

เมื่อเข้ามาในคำสั่งนี้จะเห็นว่ามีคำสั่ง 4 คำสั่ง คือ READ ,WRITE ,CLEAR และ EDIT ซึ่งเราสามารถกดคีย์ [Tab] เพื่อเลื่อนเข้าไปในคำสั่งต่างๆนี้ได้

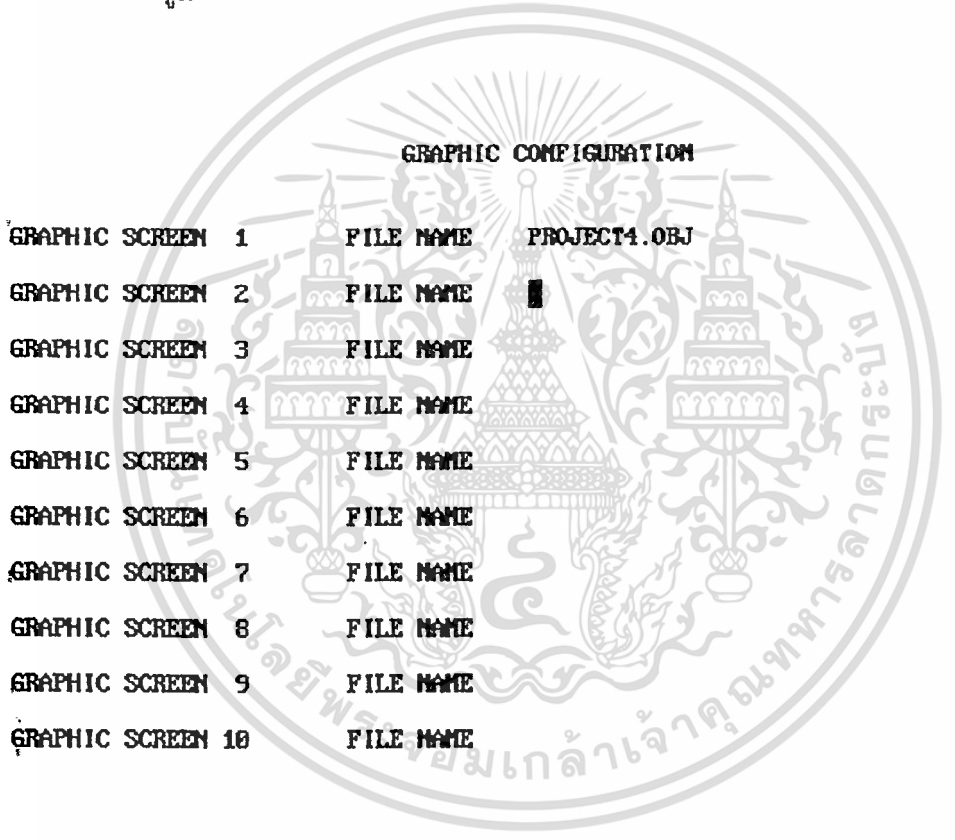
READ จะเป็นคำสั่งเพื่อเรียกข้อมูลที่เขียนไว้ขึ้นมาจากหน่วยความจำ (DCS.GRA)

WRITE จะเป็นคำสั่งเพื่อเขียนข้อมูลซึ่งเรากำหนดไว้ซึ่งเป็นข้อมูลปัจจุบันที่ปรากฏอยู่ใน EDIT ลงในหน่วยความจำ ซึ่งเป็นไฟล์ชื่อ DCS.GRA

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CLEAR จะลบข้อมูลที่กำหนดไว้ใน EDIT ขณะนั้นซึ่งจะทำให้ที่ทุก GRAPHIC SCREEN ว่างเปล่า ไม่มีไฟล์ที่กำหนดไว้เหลืออยู่

EDIT ในคำสั่ง EDIT นี้เราสามารถกำหนดและแก้ไขชื่อไฟล์ซึ่งกำหนดไว้ ในแต่ละ GRAPHIC SCREEN ได้ ซึ่งชื่อไฟล์จะต้องเป็นชื่อไฟล์ที่ตรงกับที่เรากำหนดไว้ในโปรแกรม Gedit ด้วย เช่นถ้าต้องการให้ปรากฏภาพจำลองดังรูป ให้พิมพ์ชื่อ project4.obj ที่ GRAPHIC SCREEN 1 ดังแสดงในรูป



รูปที่ 3.8 แสดงการกำหนดภาพจำลองใน EDIT ของ Graphic Display Generator

11. Group Display

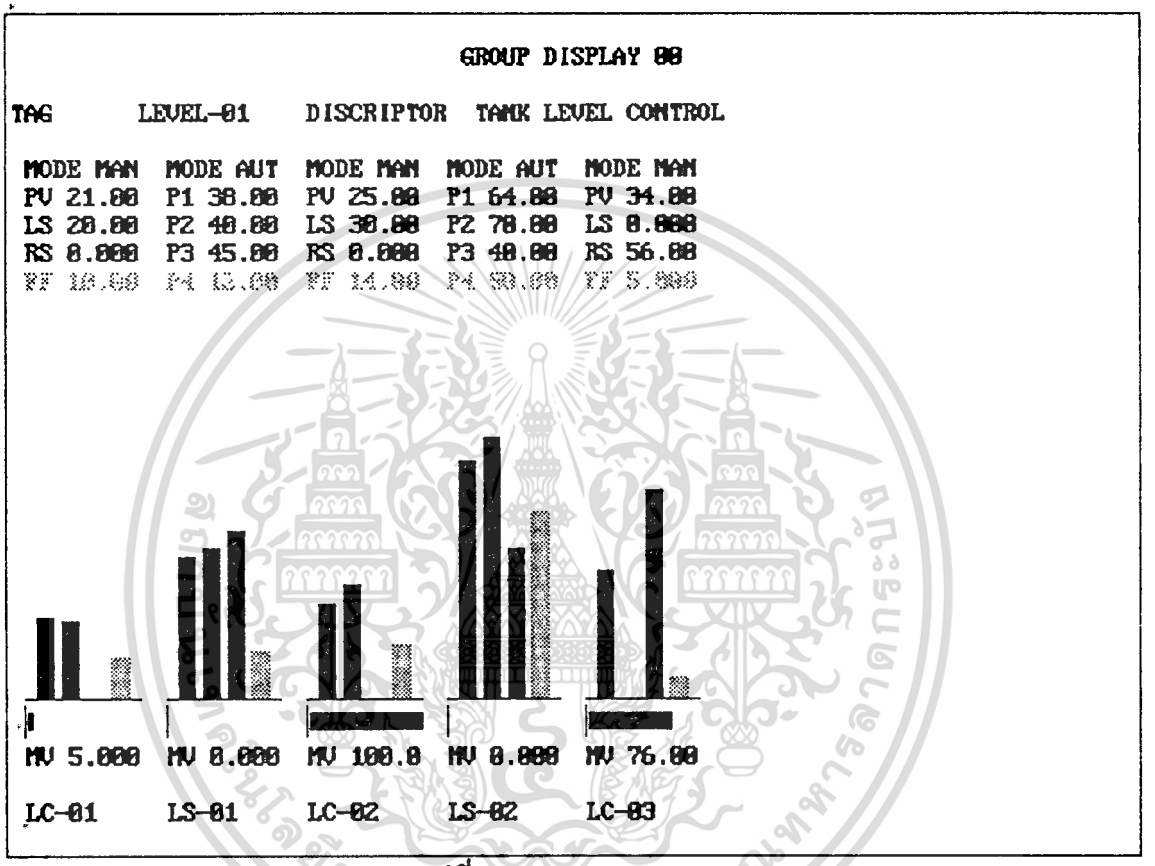
Group Display คือการนำControl Block Faceplate มาแสดงรวมกันไว้เป็นหมู่ เพื่อทำให้สามารถสังเกตุลักษณะของกระบวนการหลายกระบวนการไปพร้อมกันได้

เราจะสามารถแสดง Faceplate เป็นกลุ่ม 16 กลุ่ม คือ ตั้งแต่ GROUP DISPLAY 00 - GROUP DISPLAL 15 ซึ่งเปลี่ยนกลุ่มได้โดยกดคีย์ [Pg Up] และ [Pg Dn]

เมื่อเรากำหนด ชื่อ TAG และ DESCRIPTOR ของ GROUP DISPLAY นั้นๆ เรียบร้อยแล้ว ก็ให้ใส่ชื่อ TAG ของ Control Block Faceplate ที่เราต้องการนำมาแสดงที่แถบแดง ด้านล่าง เมื่อ

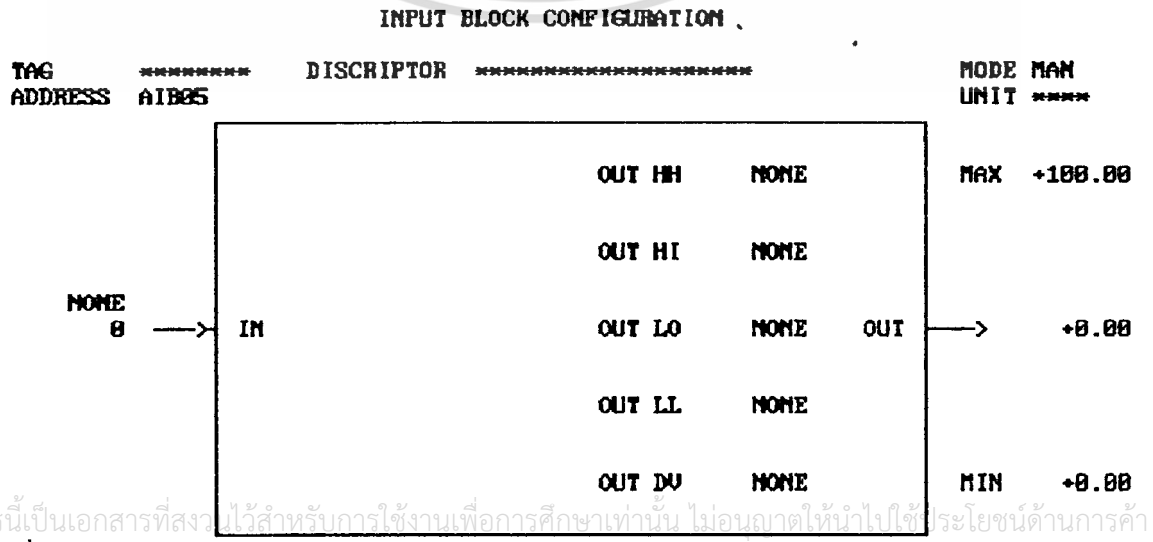
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใส่ชื่อTAG แล้วกด [Enter] ก็จะปรากฏ Faceplate ของคอนโทรลบล็อคนั้นๆ ขึ้นมา ซึ่งGroup Display หนึ่งจะสามารถแสดงได้ 8 Faceplate



รูปที่ 3.9 Group Display

12. Input Block Configuration



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.10 Input Block Configuration

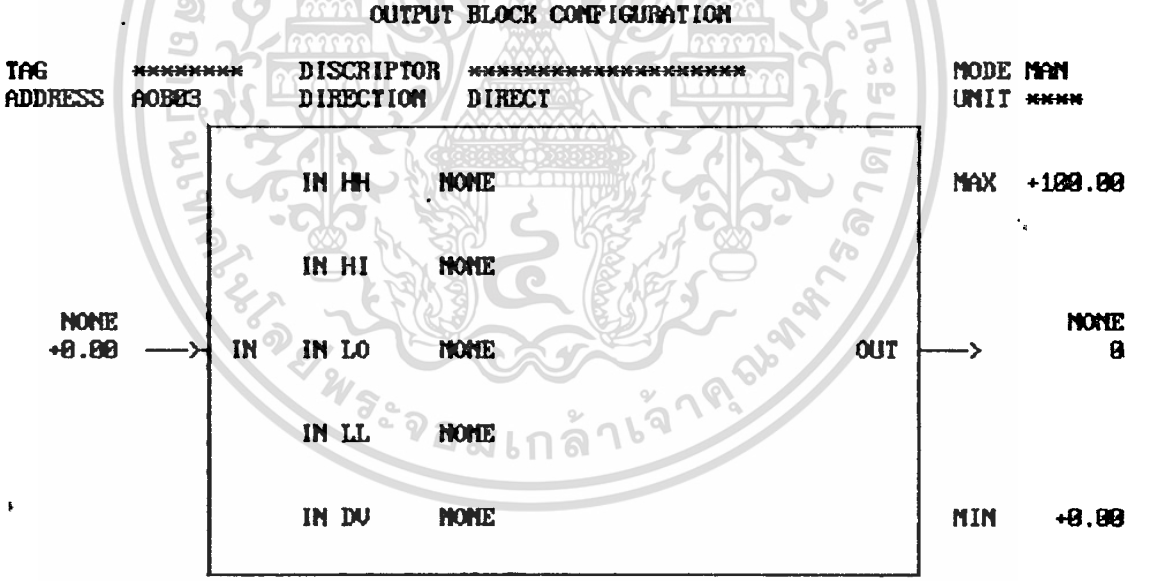
Input Block Configuration เป็นบล็อคที่ใช้รับข้อมูล ซึ่งส่งมาจากตัววัดสัญญาณในกระบวนการ เพื่อส่งต่อไปที่ Control Block Diagram ซึ่งค่าแอดเดรสของบล็อคอินพุทนี้ จะมีค่าตั้งแต่ AIB00 ถึง AIB99

ค่าอินพุต ที่ใส่เข้ามาคือ I00 ถึง I63 ซึ่งเป็นค่าอินพุตที่ได้ติดตั้งในโปรแกรม INSTALL.EXE ไว้เพื่อให้สามารถรับค่าอินพุตจากทรานสมิทเตอร์ (Transmitter) ต่างๆซึ่งส่งผ่านอินเตอร์เฟสบอร์ดมา ได้ตรงตาม channel ของอินเตอร์เฟสบอร์ด

13. Initial System Memory (เป็นคำสั่งที่ไม่ได้ใช้ในการทดลองนี้)

14. Output Block Configuration

เป็นบล็อคที่ใช้กำหนดการนำค่าเอาต์พุตจากคอนโทรลเลอร์ เพื่อส่งไปควบคุมอุปกรณ์ควบคุมสุดท้าย (Final Control Element)



รูปที่ 3.11 **Output Block Configuration**

ค่าแอดเดรสของ เอาต์พุตบล็อค จะมีตั้งแต่ AOB00 ถึง AOB99

ค่าที่รับเข้ามาเป็นอินพุตของเอาต์พุตบล็อค คือค่าเอาต์พุตที่ได้จากคอนโทรลบล็อคทำได้โดยใส่ชื่อ TAG หรือแอดเดรสของคอนโทรลบล็อคนั้นที่ INPUT ของเอาต์พุตบล็อคที่เอาต์พุต ของเอาต์พุตบล็อคจะเป็นการกำหนด channel ของอินเตอร์เฟสบอร์ด ที่จะส่งเอาต์พุตออกไปผ่านอินเตอร์เฟสบอร์ด เพื่อส่งต่อไปยังอุปกรณ์ควบคุมสุดท้าย ซึ่งค่าที่กำหนดที่เอาต์พุตของเอาต์พุตบล็อคนี้คือค่า Q00 ถึง Q63 ซึ่งได้ติดตั้งไว้ในโปรแกรม INSTALL. EXE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DIRECTION คือลักษณะของการส่งสัญญาณไปควบคุมอุปกรณ์ควบคุมสุดท้าย มี 2 ลักษณะคือ Direct และ Reverse ซึ่งกำหนดได้โดยกดคีย์ [Pg Up] และ [Pg Dn]

15. Run Mode

Run Mode เป็นโหมดที่ใช้ในการปฏิบัติการ เมื่ออยู่ใน Run Mode คอนโทรลเลอร์จะทำงานและสามารถติดต่อกับอุปกรณ์ภายนอกได้ ซึ่งเมื่อเข้า Run Mode จะทำให้เกิดการปฏิบัติการขึ้นทันที จึงควรแน่ใจว่ากำหนดตัวแปรต่างๆ ใน configuration ต่าง ถูกต้องแล้ว เมื่ออยู่ใน Run Mode บล็อคการทำงานต่าง ๆ จะมีวันที่ (Date) ทางมุมบนซ้ายและเวลา (Time) ทางมุมบนขวา

16. Read System Memory

เป็นคำสั่งที่ใช้เพื่ออ่านค่าของข้อมูลต่างๆ จาก System memory คือ ไฟล์ชื่อ. DCS.BLK ขึ้นมาในบล็อคต่าง ๆ

17. System Configuration

System Configuration เป็นบล็อคที่นำเอา Configuration ของระบบทั้งหมดมารวมไว้ เพื่อให้ง่ายต่อการจัดการโดยรวม

```
                    SYSTEM CONFIGURATION

SYSTEM CONFIGURATION      FILE NAME  DCS.SYS
SYSTEM MEMORY             FILE NAME  DCS.BLK
SEQUENTIAL PROGRAM        FILE NAME
GRAPHIC CONFIGURATION     FILE NAME  DCS.GRA
GROUP CONFIGURATION       FILE NAME  DCS.GRP
TREND CONFIGURATION       FILE NAME  DCS.TRD
ALARM LIST                FILE NAME
```

รูปที่ 3.12 **System Configuration**

System Memory จะมีคำสั่งซึ่งกระทำกับ System Memory ซึ่งคือไฟล์ชื่อ DCS.BLK ซึ่งคำสั่งมีดังนี้.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    SYSTEM MEMORY CONFIGURATION

SYSTEM MEMORY          FILE NAME  DCS.BLK
                                  READ
                                  WRITE
                                  CLEAR
```

รูปที่ 3.13 System Memory Configuration

- READ เป็นคำสั่งที่ใช้โหลดข้อมูลจากไฟล์ DCS.BLKขึ้นมาไว้ในบล็อคต่างๆ
- WRITE เป็นคำสั่งที่ใช้เขียนข้อมูล ที่อยู่ในบล็อคขณะนั้นลงไปเก็บไว้ในไฟล์ DCS.BLK.
- CLEAR เป็นคำสั่งที่ใช้เคลียร์ให้บล็อคขณะนั้นว่างเปล่า

GRAPHIC CONFIGURATION เป็นคำสั่งซึ่งใช้ในการกำหนดค่าต่างๆ ของ Graphic Display ซึ่งจะเหมือนกับคำสั่งใน Graphic Display Generator

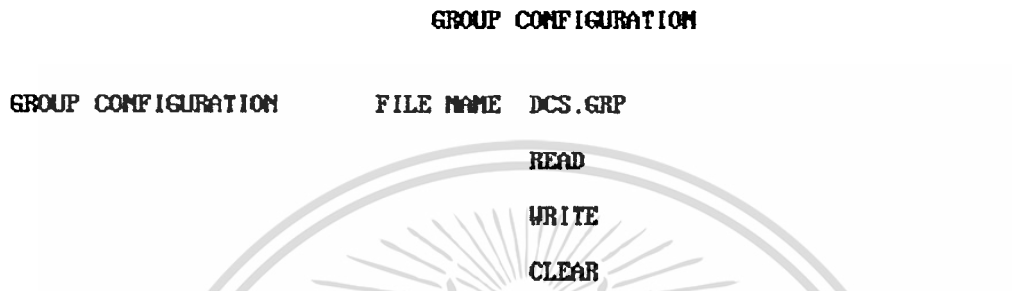
```
                    GRAPHIC CONFIGURATION

GRAPHIC CONFIGURATION     FILE NAME  DCS.GRA
                                     READ
                                     WRITE
                                     CLEAR
                                     EDIT
```

รูปที่ 3.14 Graphic Configuration ใน System Configuration

- READ เป็นคำสั่งที่ใช้โหลดข้อมูลจากไฟล์ DCS.GRAขึ้นมาไว้ในบล็อคEDIT ของ Graphic Display Generator และบล็อค Graphic Display
- WRITE เป็นคำสั่งที่ใช้เขียนข้อมูล ที่อยู่ในบล็อค Graphic Display Generator ขณะนั้นลงไปเก็บไว้ในไฟล์ DCS.GRA
- CLEAR เป็นคำสั่งที่ใช้เคลียร์ให้บล็อคEDIT ของ Graphic Display Generator และบล็อค Graphic Display ขณะนั้นว่างเปล่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- EDIT ในคำสั่ง EDIT นี้เราสามารถกำหนดและแก้ไขชื่อไฟล์ซึ่งกำหนดไว้ ในแต่ละ GRAPHIC SCREEN ได้ ซึ่งชื่อไฟล์จะต้องเป็นชื่อไฟล์ที่ตรงกับที่เรากำหนดไว้ในโปรแกรม Gedit ด้วย



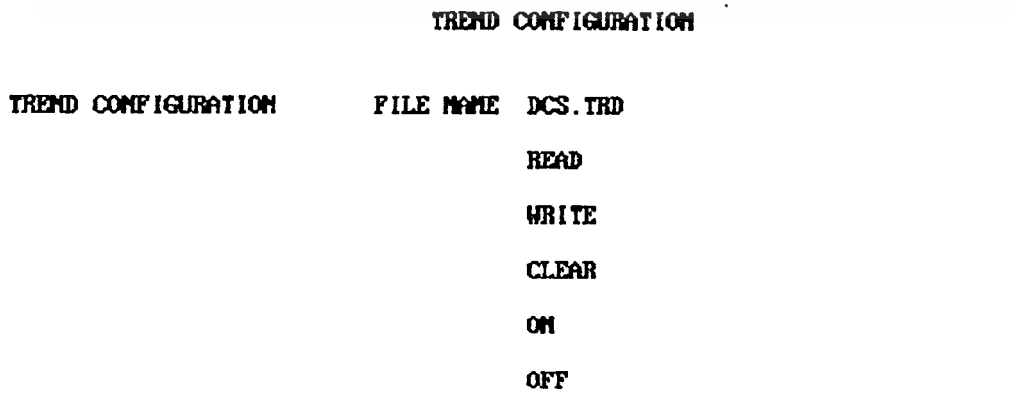
รูปที่ 3.15 Group Configuration ใน System Configuration

GROUP CONFIGURATION เป็นคำสั่งที่ใช้ในการกำหนดการเขียนอ่านข้อมูลในบล็อคของ GRoup Display กับไฟล์ ชื่อ DCS.GRP ซึ่งมีคำสั่งดังนี้. READ เป็นคำสั่งที่ใช้โหลดข้อมูลจากไฟล์ DCS.GRP ขึ้นมาสู่บล็อคของ GRoup Display

- WRITE เป็นคำสั่งที่ใช้เขียนข้อมูลที่มีอยู่ในบล็อค GRoup Display ลงไปเก็บไว้ในไฟล์ DCS.GRP

- CLEAR เป็นคำสั่งที่ใช้ เคลียร์ให้บล็อค GRoup Display เป็นบล็อคว่างเปล่า.

TREND CONFIGURATION . เป็นคำสั่งที่ใชในการกำหนดการเขียนอ่านข้อมูลในบล็อคของ Tranding Display กับไฟล์ ชื่อ DCS.TRD และกำหนดการเปิดและปิด ของ Trending Display ซึ่งมีคำสั่งดังนี้



รูปที่ 3.16 Trend Configuration ใน System Configuration

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- READ เป็นคำสั่งที่ใช้โหลดข้อมูลจากไฟล์ DCS.TRD ขึ้นมาสู่บล็อคของ Trending Display
- WRITE เป็นคำสั่งที่ใช้เขียนข้อมูลที่มีอยู่ในบล็อค Trending Display ลงไปเก็บไว้ในไฟล์ DCS.TRD
- CLEAR เป็นคำสั่งที่ใช้เคลียร์ให้บล็อค Trending Display เป็นบล็อคว่างเปล่า.
- ON เป็นคำสั่งที่ใช้การทำงานของ Trending Display ซึ่งทำให้ Trending Dusokat เข้าสู่ระบบปฏิบัติการ ในขณะนี้จะไม่สามารถเข้าไปกำหนดตัวแปรต่างๆในส่วนปฏิบัติการของ Trending Display ได้.
- OFF เป็นคำสั่งที่ใช้ปิดการทำงานของ Trending Display ซึ่งทำให้สามารถเข้าไปกำหนด

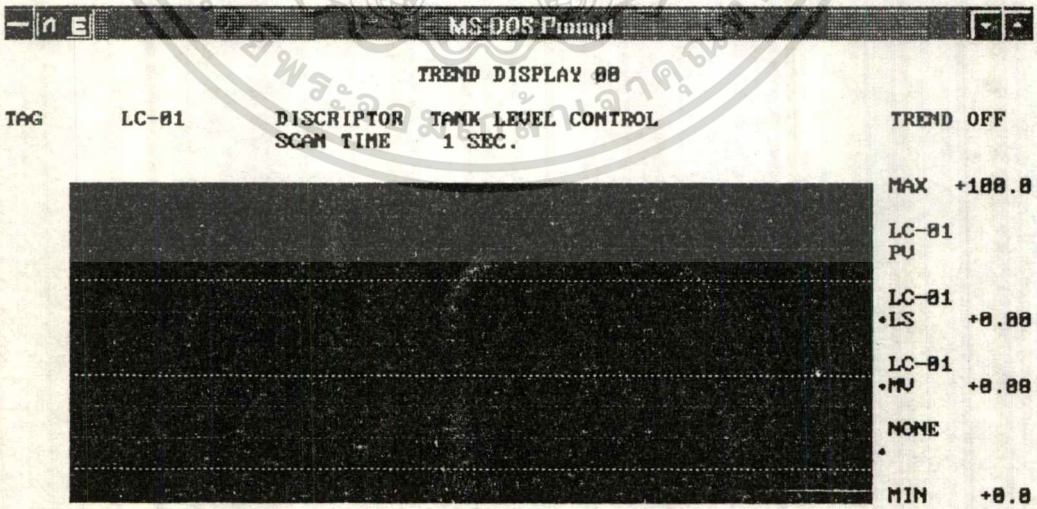
18. Sequential Control Monitor. (เป็นคำสั่งที่ไม่ได้ใช้ในการทดลองนี้)

19. Sequential Control Program. (เป็นคำสั่งที่ไม่ได้ใช้ในการทดลองนี้)

20. Standby Mode

เป็นโหมดซึ่งไม่มีการปฏิบัติการใด ๆ เกิดขึ้นในโหมดนี้ เมื่อเข้ามาในโปรแกรมตอนแรกจะอยู่ในโหมดนี้ และก่อนที่จะออกจากโปรแกรมก็ต้องเข้ามาในโหมดนี้ก่อน

21. Trending Display.



รูปที่ 3.17 Trending Display

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็นส่วนที่ใช้แสดงการเปลี่ยนแปลงของกระบวนการโดยใช้กราฟโดยสามารถแสดงได้ 16 Display คือ. TREND DISPLAY 00 ถึง TREND DISPLAY 15 โดยสามารถเปลี่ยนไปดู TREND DISPLAY แต่ละบล็อคได้โดยกดคีย์ [Pg Up] และ [Pg Dn]

MAX คือค่าสูงสุดที่ต้องการแสดง

MIN คือค่าต่ำสุดที่ต้องการแสดง

ในการนำค่าของกระบวนการมาแสดงทำได้โดยกำหนด ชื่อ TAG ของคอนโทรลบล็อคที่เราต้องการนำมาแสดง แล้วกดคีย์ [Enter] หลังจากนั้นจะมีแถบแดงขึ้นที่ด้านล่างของชื่อ TAG สามารถเลือกค่าหรือตัวแปรที่ต้องการให้มันแสดงโดยกดคีย์ [Pg Up] และ [Pg Dn] เมื่อตัวแปรที่ต้องการปรากฏอยู่จึงกด[Enter]

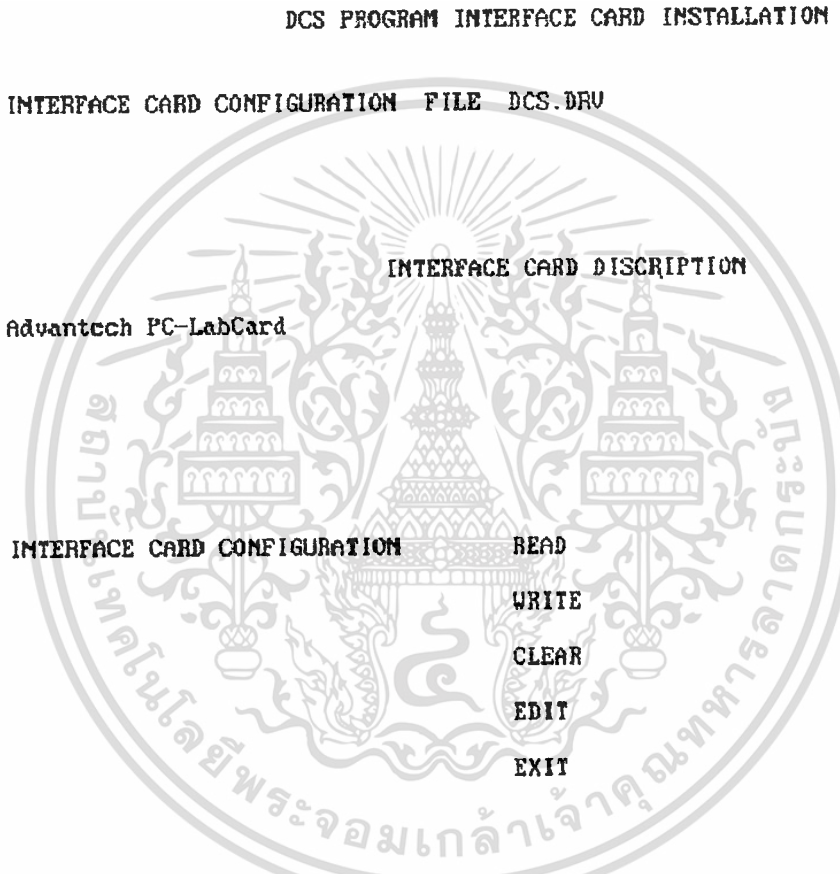
22. Write System Memory

เป็นคำสั่งที่ใช้เขียนข้องมูลที่มีอยู่ในบล็อคต่าง ๆ ขณะนั้นจะไปเก็บ System Memory คือ ไฟล์ DCS.BLK

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การเชื่อมต่อโปรแกรม DCS กับการ์ดแปลงสัญญาณ (A/D , D/A)

การเชื่อมต่อโปรแกรม DCS กับการ์ดแปลงสัญญาณ จะเชื่อมต่อโดยใช้โปรแกรม INSTALL เมื่อเข้ามาในโปรแกรมนี้แล้วจะปรากฏหน้าจอดังรูป



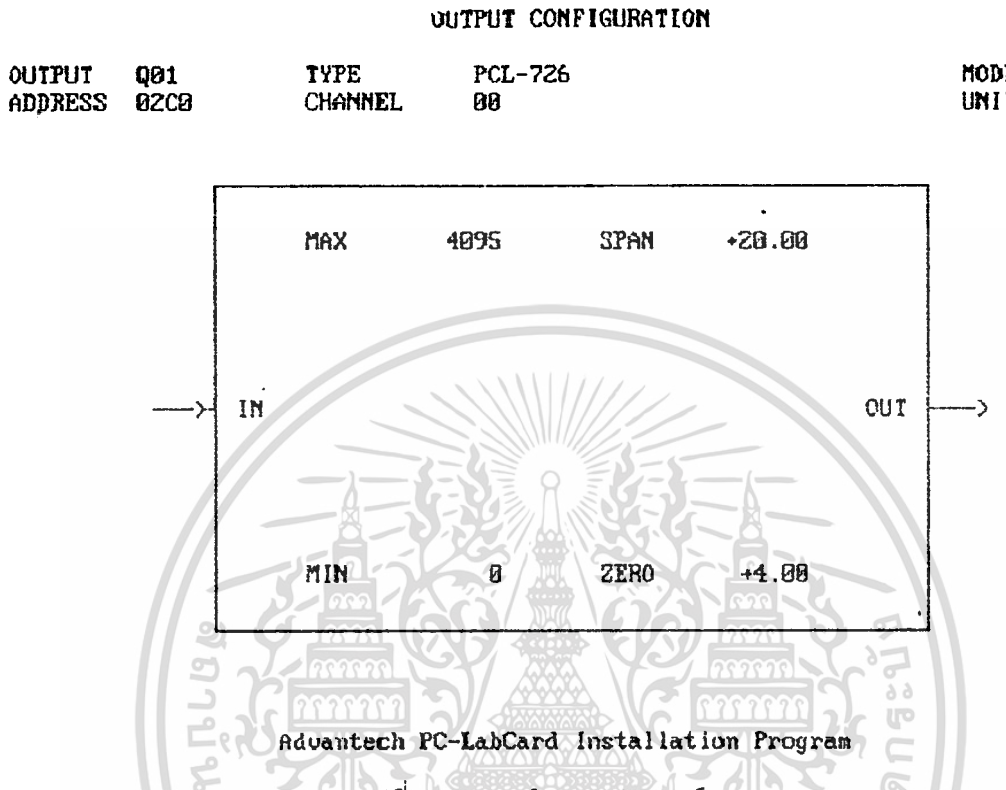
รูปที่ 3.18 หน้าจอของโปรแกรม Installation

ข้อมูลที่ใช้ในการ configuration จะถูกเก็บในไฟล์ชื่อ DCS.DRV คำสั่งที่ใช้ในการ configuration มีดังนี้

- READ เป็นคำสั่งที่ใช้โหลดข้อมูลจากไฟล์ DCS.DRVขึ้นมาไว้ในบล็อคEDIT
- WRITE เป็นคำสั่งที่ใช้เขียนข้อมูล ที่อยู่ในบล็อค EDIT ขณะนั้นลงไปเก็บไว้ในไฟล์ DCS.DRV
- CLEAR เป็นคำสั่งที่ใช้เคลียร์ให้บล็อค EDIT ขณะนั้นว่างเปล่า
- EDIT ในคำสั่ง EDIT นี้เราสามารถกำหนดการติดต่อระหว่างการ์ดกับโปรแกรมได้ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้ ห้ามมิให้ทำซ้ำ ดัดแปลง เผยแพร่ หรือจำหน่ายในรูปแบบใดๆ ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การกำหนดค่าสัญญาณออก (OUTPUT CONFIGURATION)



รูปที่3.19 การกำหนดค่าเอาต์พุต

OUTPUT คือชื่อที่ใช้ติดต่อระหว่างแชนแนล (channel) ของเอาต์พุตของการ์ดแปลงสัญญาณดิจิตอลเป็นสัญญาณอนาล็อกกับสัญญาณจากเอาต์พุตบล็อคของโปรแกรม DCS

TYPE คือชื่อของการ์ดแปลงสัญญาณดิจิตอลอนาล็อกเป็นสัญญาณอนาล็อกที่นำมาใช้

MODE มี 2 โหมดคือ

ANALOG ส่งสัญญาณออกเป็นสัญญาณอนาล็อก

DIGITAL ส่งสัญญาณออกเป็นสัญญาณดิจิตอล

ADDRESS คือแอดเดรสในหน่วยความจำของคอมพิวเตอร์ที่การ์ดใช้ในการติดต่อซึ่งโดยปกติแล้วผู้ผลิตการ์ดจะกำหนดมา

CHANNEL คือแชนแนลของการ์ดที่ใช้ติดต่อ

UNIT คือหน่วยของสัญญาณที่ส่งออกไป

MAX คือค่าของสัญญาณออกสูงสุดเป็นเลขฐานสิบหก

MIN คือค่าของสัญญาณออกต่ำสุดเป็นเลขฐานสิบหก

SPAN คือค่าของสัญญาณออกสูงสุดเป็นเลขฐานสิบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

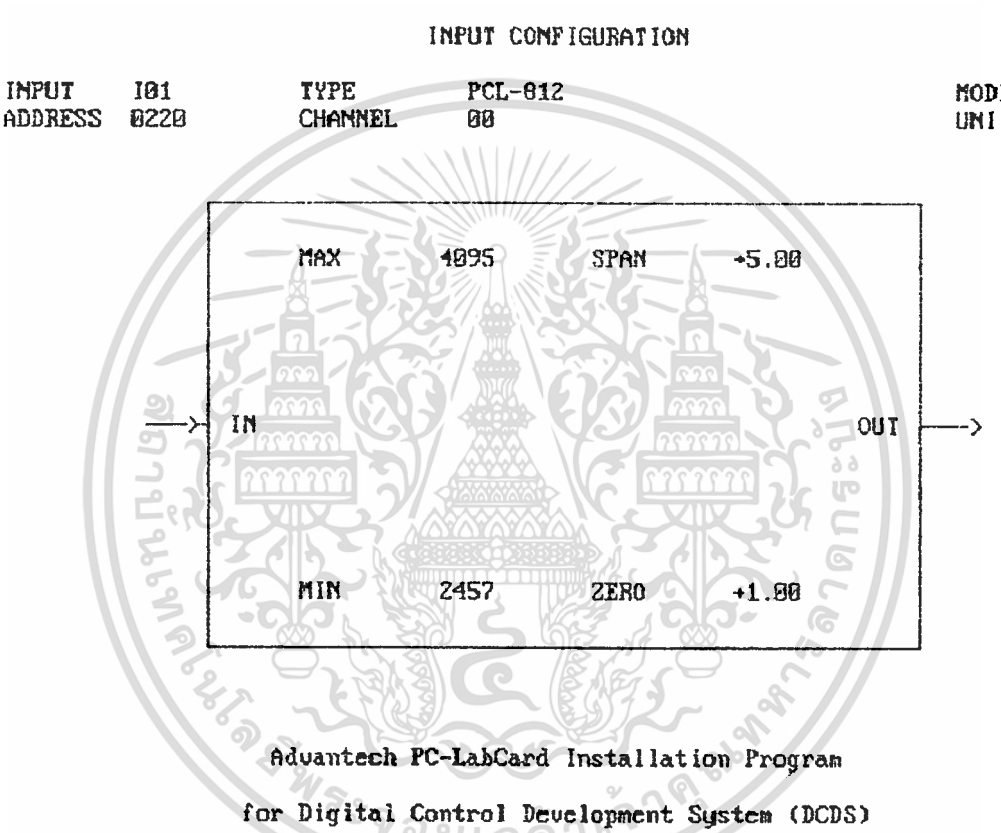
ZERO คือค่าของสัญญาณออกต่ำสุดเป็นเลขฐานสิบ

สำหรับการทดลองนี้ กำหนดค่าอินพุตดังนี้

เอาต์พุต Q01 ติดต่อกับการ์ดPCL-726 แชนแนล00 โดยส่งสัญญาณออกเป็น 4ถึง 20 mA

เอาต์พุต Q02 ติดต่อกับการ์ดPCL-726 แชนแนล01 โดยส่งสัญญาณออกเป็น 4ถึง 20 mA

การกำหนดค่าสัญญาณเข้า (INPUT CONFIGURATION)



รูปที่ 3.20 การกำหนดค่าอินพุต

INPUT คือชื่อที่ใช้ติดต่อระหว่างแชนแนล (channel) ของอินพุตของการ์ดแปลงสัญญาณอนาล็อกเป็นสัญญาณดิจิตอลกับสัญญาณจากเอาต์พุตบล็อคของโปรแกรม DCS

TYPE คือชื่อของการ์ดแปลงสัญญาณอนาล็อกเป็นสัญญาณดิจิตอลที่นำมาใช้

MODE มี 2 โหมดคือ

ANALOG ส่งสัญญาณออกเป็นสัญญาณอนาล็อก

DIGITAL ส่งสัญญาณออกเป็นสัญญาณดิจิตอล

ADDRESS คือแอดเดรสในหน่วยความจำของคอมพิวเตอร์ที่การ์ดใช้ในการติดต่อซึ่งโดยปกติแล้วผู้ผลิตการ์ดจะกำหนดมา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CHANNEL คือแชนแนลของการ์ดที่ใช้ติดต่อ

UNIT คือหน่วยของสัญญาณที่รับเข้ามา

MAX คือค่าของสัญญาณเข้าสูงสุดเป็นเลขฐานสิบหก

MIN คือค่าของสัญญาณเข้าต่ำสุดเป็นเลขฐานสิบหก

SPAN คือค่าของสัญญาณเข้าสูงสุดเป็นเลขฐานสิบ

ZERO คือค่าของสัญญาณเข้าต่ำสุดเป็นเลขฐานสิบ

สำหรับการทดลองนี้ กำหนดค่าอินพุตดังนี้

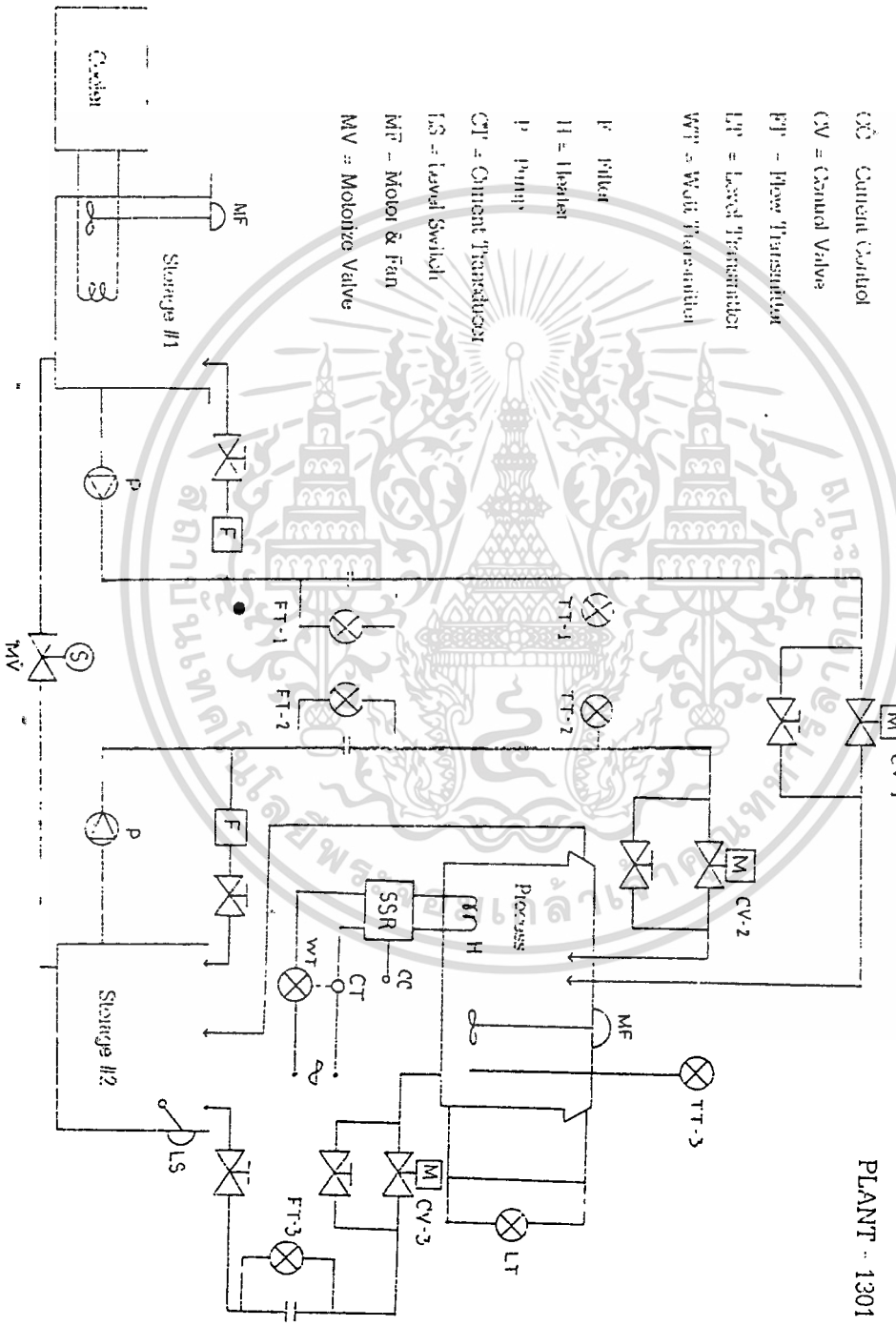
อินพุต I01 ติดต่อกับการ์ด PCL-812 แชนแนล 00 โดยรับสัญญาณเข้าเป็น 1ถึง 5 VDC

อินพุต I02 ติดต่อกับการ์ด PCL-812แชนแนล 01 โดยรับสัญญาณเข้าเป็น 1ถึง 5 VDC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

2. กระบวนการ

ในการทดลองใช้ Plant-1301 ซึ่งมีบล็อคไดอะแกรมดังรูป



รูปที่ 3.21 Plant-1301

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. ส่วนเชื่อมต่อ (Interface Board)

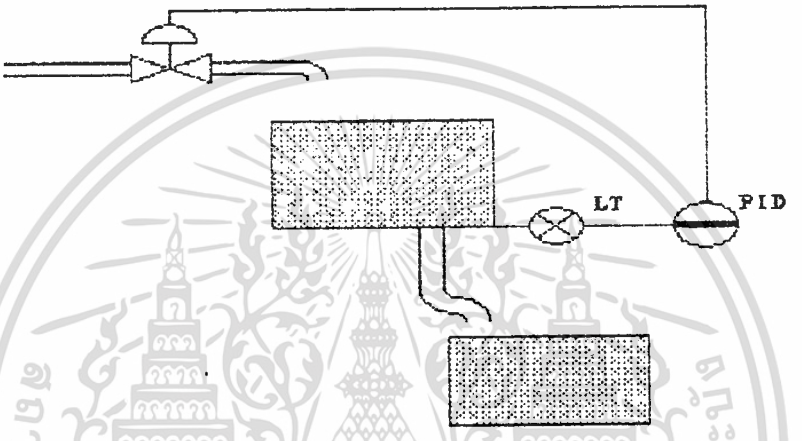
การติดต่อระหว่างกระบวนการกับโปรแกรม DCS ใช้การ์ด PCL-812 เป็นตัวแปลงสัญญาณอนาล็อกเป็นดิจิตอล และใช้การ์ด PCL-726 เป็นตัวแปลงสัญญาณดิจิตอลเป็นอนาล็อก ซึ่งรายละเอียดของการ์ดทั้ง 2 มีอยู่ในภาคผนวก ส่วนการเชื่อมโยงระหว่างกระบวนการกับการ์ดแปลงสัญญาณจะใช้บอร์ดเชื่อมต่อ PCL-780 สองบอร์ด PCL-780(A) ติดต่อกับการ์ด PCL-726 และPCL-780(B) ติดต่อกับการ์ด PCL-812 ระหว่างบอร์ดPCL-780 กับการ์ดแต่ละตัวจะเชื่อมต่อด้วยสายแพ ขนาด 20 เส้น

สำหรับสัญญาณที่ส่งออกมาจากทรานสมิตเตอร์ของการกระบวนการจะเป็นสัญญาณ 4-20 mA แต่การ์ด PCL-812(A/D) รับสัญญาณเข้าเป็น 1 - 5 โวลต์ ดังนั้นที่บอร์ด PCL-780(B) จะมีความต้านทานขนาด 250 โอห์มต่ออยู่ ที่ 8 แชนเนลแรก คือแชนเนล 0 - 7

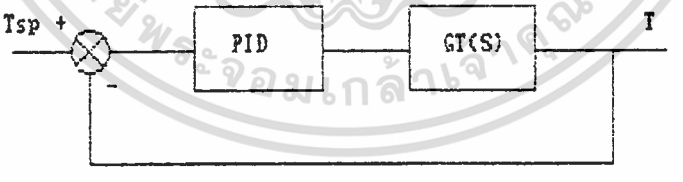
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
การดำเนินงาน

1. การควบคุมระดับน้ำ (LEVEL CONTROL)



รูป 4.1 การควบคุมระดับน้ำ (LEVEL CONTROL)



รูป 4.2 บล็อคไดอะแกรมแสดงการควบคุมระดับน้ำ

ขั้นตอนการดำเนินงาน

1. ต่อสายสัญญาณจาก LEVEL TRANSMITTER มาเข้าที่ channel 0 ของบอร์ดPCLD 780 (B)
2. กำหนดค่าใน Input Block Configuration เลือกใช้แอดเดรส AIB00 กำหนดค่าอินพุต (IN) เป็น I01

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กำหนดค่าต่ำสุดของเอาต์พุต (MIN) เป็น 5

กำหนดค่าสูงสุดของเอาต์พุต (MAX) เป็น 24

กำหนดหน่วย (UNIT) เป็นเซนติเมตร (CM)

3. กำหนดค่าใน Control Block Diagram เลือกใช้แอดเดรส ACB00

กำหนด TAG เป็น LC

เลือกฟังก์ชันเป็น PID Control

กำหนดตัวแปรกระบวนการ PV เป็นค่าแอดเดรสของ IBC คือ AIB00

กำหนดเซตพอยน์ที่ LS เป็นค่าที่เราต้องการ

กำหนดค่าต่ำสุดของเอาต์พุต (MIN) เป็น 0

กำหนดค่าสูงสุดของเอาต์พุต (MAX) เป็น 100

กำหนดหน่วย (UNIT) เป็นเปอร์เซนต์ (%)

4. กำหนดค่าใน Output Block Configuration เลือกใช้แอดเดรส AOB00

กำหนด TAG เป็น LV

กำหนดทิศทางการบังคับวาล์ว (DIRECTION) เป็น DIRECT

กำหนดค่าอินพุต (IN) เป็น TAG ของ CBD คือ LC

กำหนดค่าต่ำสุดของเอาต์พุต (MIN) เป็น 0

กำหนดค่าสูงสุดของเอาต์พุต (MAX) เป็น 100

กำหนดหน่วย (UNIT) เป็นเปอร์เซนต์ของการเปิดปิดวาล์ว (%)

กำหนดค่าเอาต์พุต (OUT) เป็น Q01

5. ต่อสายสัญญาณจาก channel 0 ของบอร์ดPCLD 780 (A) มาเข้าที่ control valve (CV) ของกระบวนการ

6. กำหนดค่าพารามิเตอร์ของตัวควบคุมที่ Control Block Faceplate

7. กำหนดโหมดการทำงานใน IBC, OBC, CBD เป็นอัตโนมัติ (AUT)

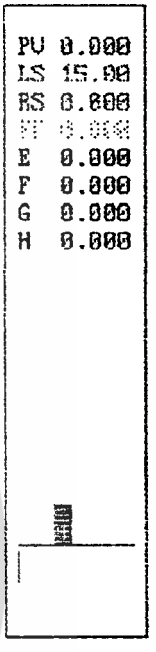
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CONTROL BLOCK FACEPLATE

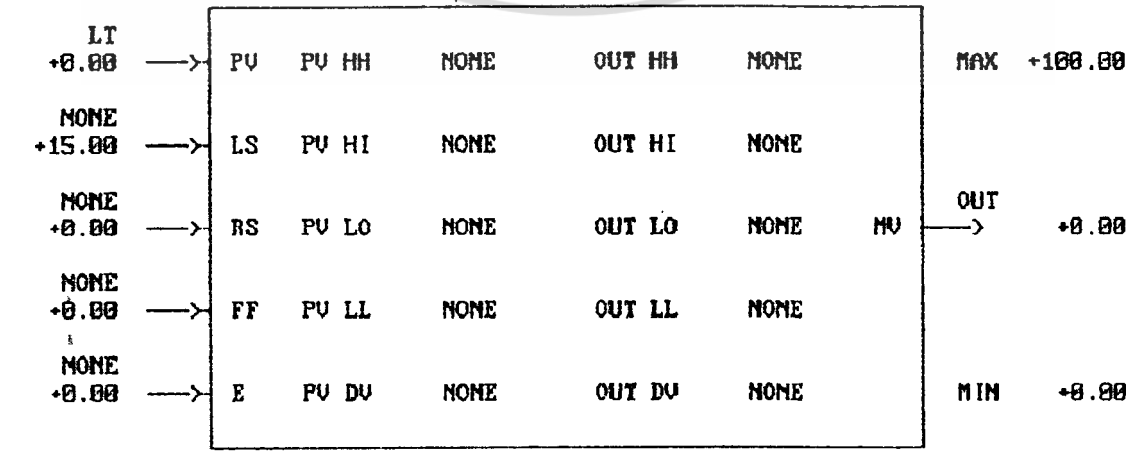
```
TAG       LC           DISCRIPTOR  TANK LEVEL CONTROL          MODE AUT
ADDRESS   ACB00        FUNCTION    PID Control

PV LT          +0.00          E  NONE          +0.00
LS NONE        +15.00         F  NONE          +0.00
RS NONE        +0.00          G  NONE          +0.00
FF NONE        +0.00          H  NONE          +0.00

Kp Gain        +100.00
Ti Int Time    +50.00   sec.
Id Dev Time    +0.00    sec.
Kf FF Gain     +0.00
```

OUT = Kp*(SP-PV)+Σ(Ts/Ti)*(SP-PV)+Kp*(Td/Ts)(ΔPV)+Kf*FF

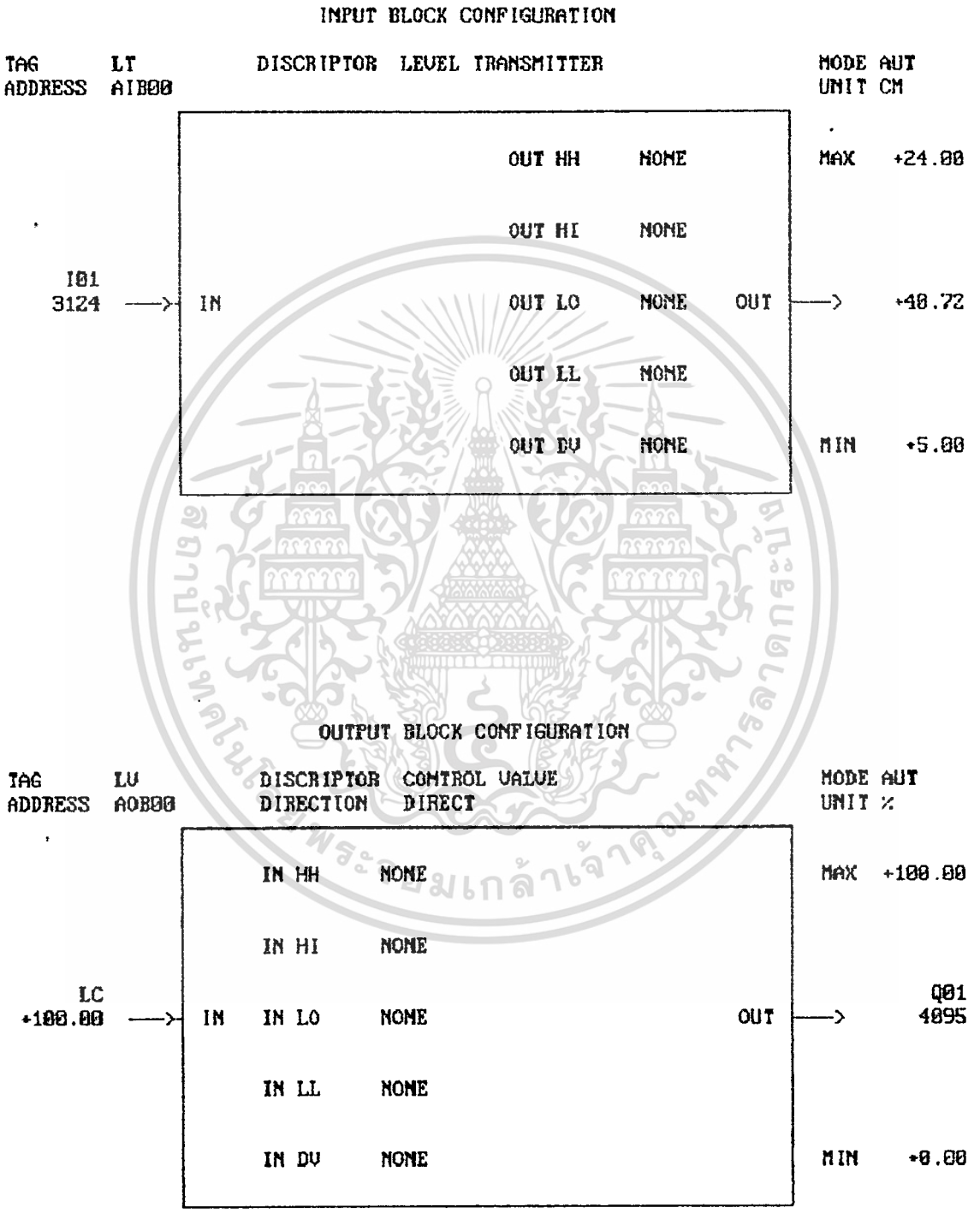
```
OUT            +0.00  %
```



CONTROL BLOCK DIAGRAM



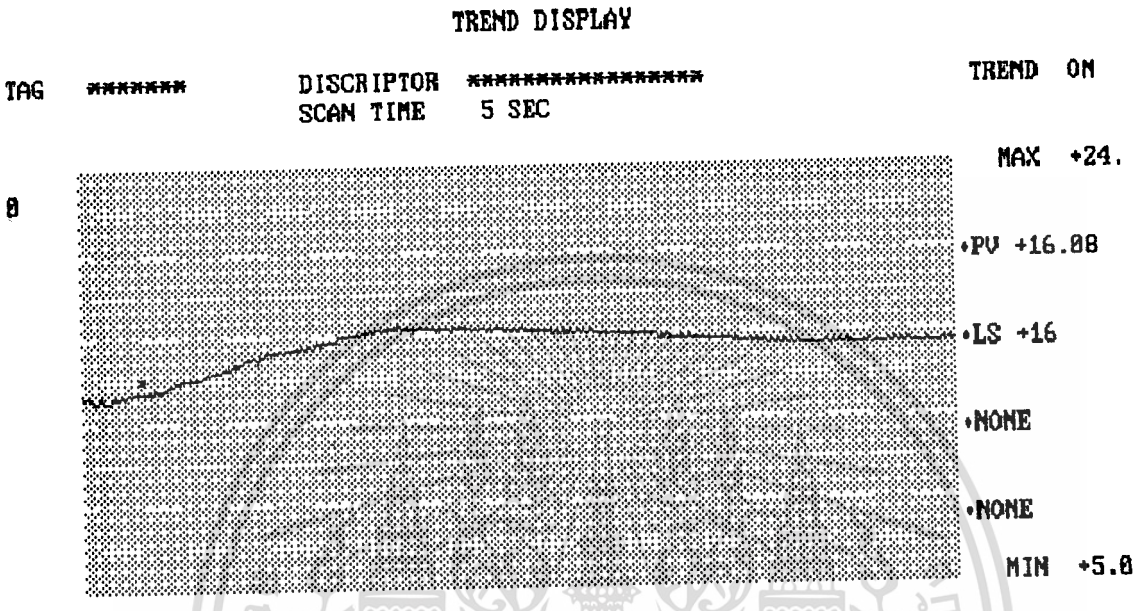
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 4.3 แสดงบล็อคการทำงานของ
โปรแกรม DCS ในการควบคุมระดับน้ำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลการทดลอง



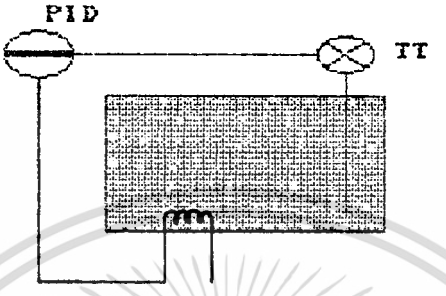
รูปที่ 4.4 รูปแสดงผลการทดลองการควบคุมระดับน้ำ

จากการทดลองได้ค่าพารามิเตอร์ของตัวควบคุมดังนี้ P=100 , I=50

จากผลการทดลองพบว่าการควบคุมระดับน้ำจะเข้าสู่สภาวะคงที่ได้เสมอ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.การควบคุมอุณหภูมิของน้ำ (TEMPERATURE CONTROL)



รูป 4.5 การควบคุมอุณหภูมิของน้ำ (TEMPERATURE CONTROL)

รูป 4.6 บล็อคไดอะแกรมแสดงการควบคุมอุณหภูมิน้ำ

ขั้นตอนการดำเนินงาน

1. ต่อสายสัญญาณจาก TEMPERATURE TRANSMITTER มาเข้าที่ channel 0 ของบอร์ดPCLD 780 (B)

2. กำหนดค่าใน Input Block Configuration เลือกใช้แอดเดรส AIB00

กำหนดค่าอินพุต (IN) เป็น I01

กำหนดค่าต่ำสุดของเอาต์พุต (MIN) เป็น 0

กำหนดค่าสูงสุดของเอาต์พุต (MAX) เป็น 100

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กำหนดหน่วย (UNIT) เป็นองศาเซลเซียส (C)

3. กำหนดค่าใน Control Block Diagram เลือกใช้แอดเดรส ACB00

กำหนด TAG เป็น TC

เลือกฟังก์ชันเป็น PID Control

กำหนดตัวแปรกระบวนการ PV เป็นค่าแอดเดรสของ IBC คือ AIB00

กำหนดเซตพอยน์ที่ LS เป็นค่าที่เราต้องการ

กำหนดค่าต่ำสุดของเอาต์พุต (MIN) เป็น 0

กำหนดค่าสูงสุดของเอาต์พุต (MAX) เป็น 100

กำหนดหน่วย (UNIT) เป็นเปอร์เซนต์ (%)

4. กำหนดค่าใน Output Block Configuration เลือกใช้แอดเดรส AOB00

กำหนด TAG เป็น TV

กำหนดทิศทางการบังคับวาล์ว (DIRECTION) เป็น DIRECT

กำหนดค่าอินพุต (IN) เป็น TAG ของ CBD คือ TC

กำหนดค่าต่ำสุดของเอาต์พุต (MIN) เป็น 0

กำหนดค่าสูงสุดของเอาต์พุต (MAX) เป็น 100

กำหนดหน่วย (UNIT) เป็นเปอร์เซนต์ของให้ความร้อน (%)

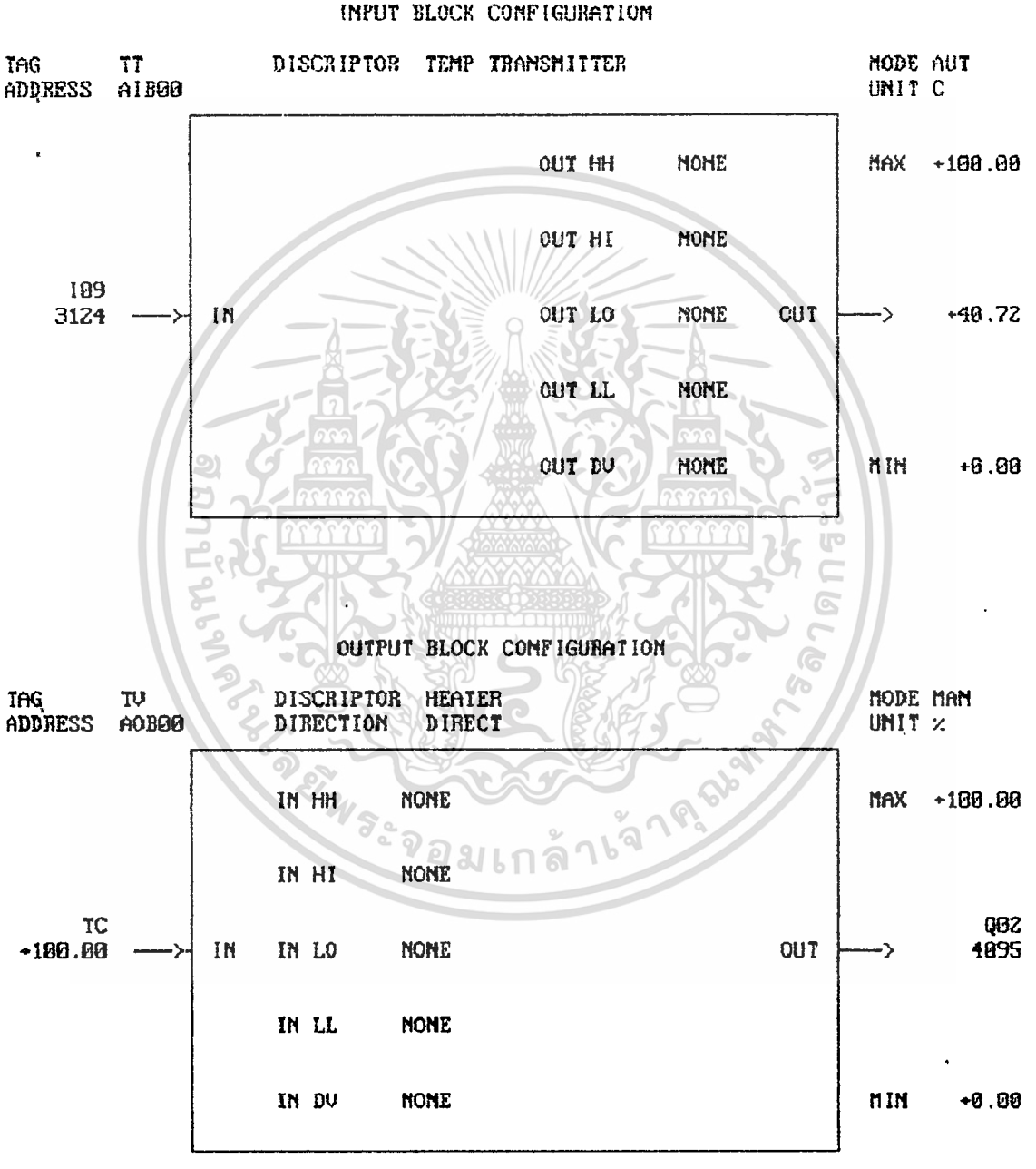
กำหนดค่าเอาต์พุต (OUT) เป็น Q01

5. ต่อสายสัญญาณจาก channel 0 ของบอร์ดPCLD 780 (A) มาเข้าที่ Watt Transducer (WT) ของกระบวนการ

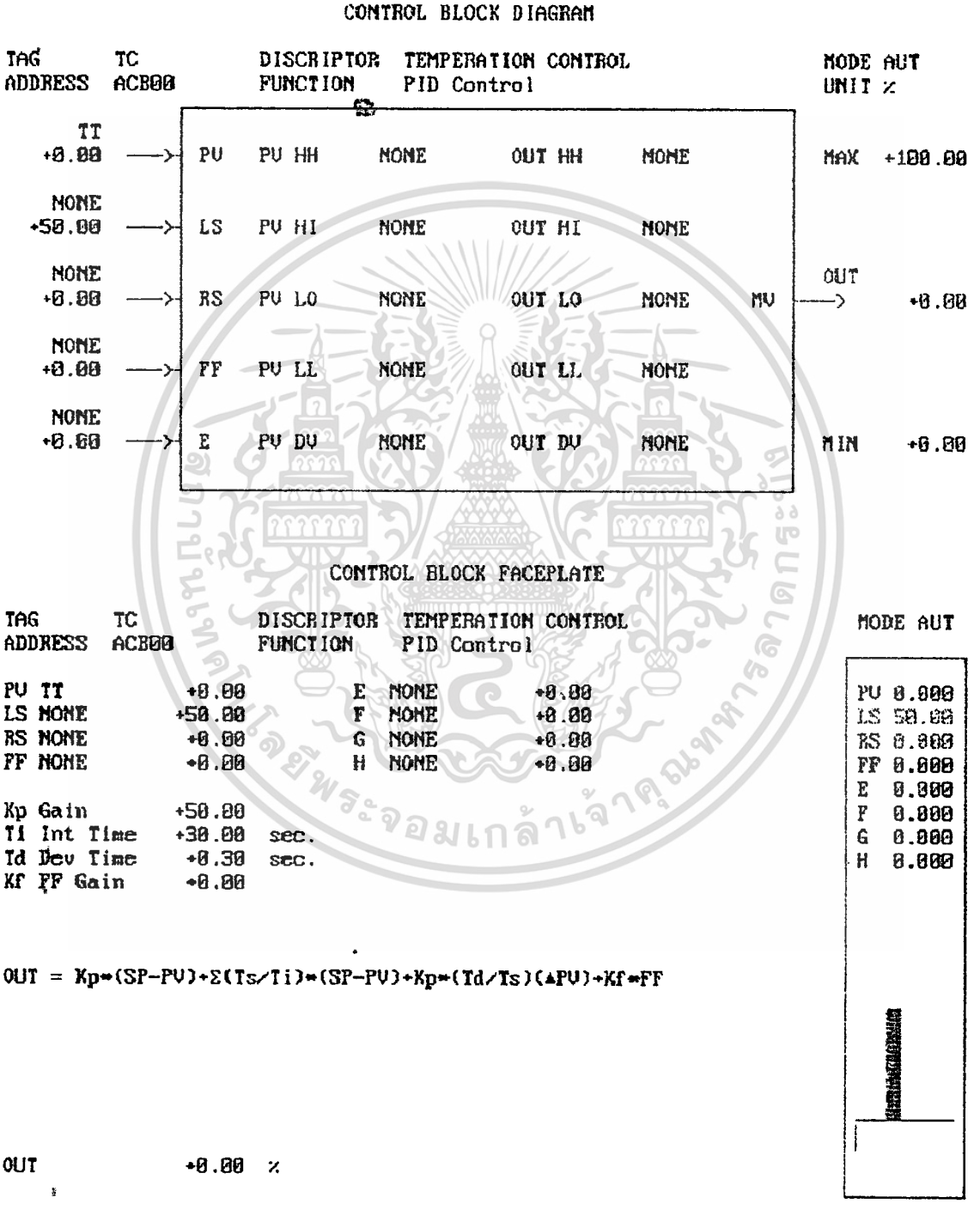
6. กำหนดค่าพารามิเตอร์ของตัวควบคุมที่ Control Block Faceplate

7. กำหนดโหมดการทำงานใน IBC, OBC, CBD เป็นอัตโนมัติ (AUT)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



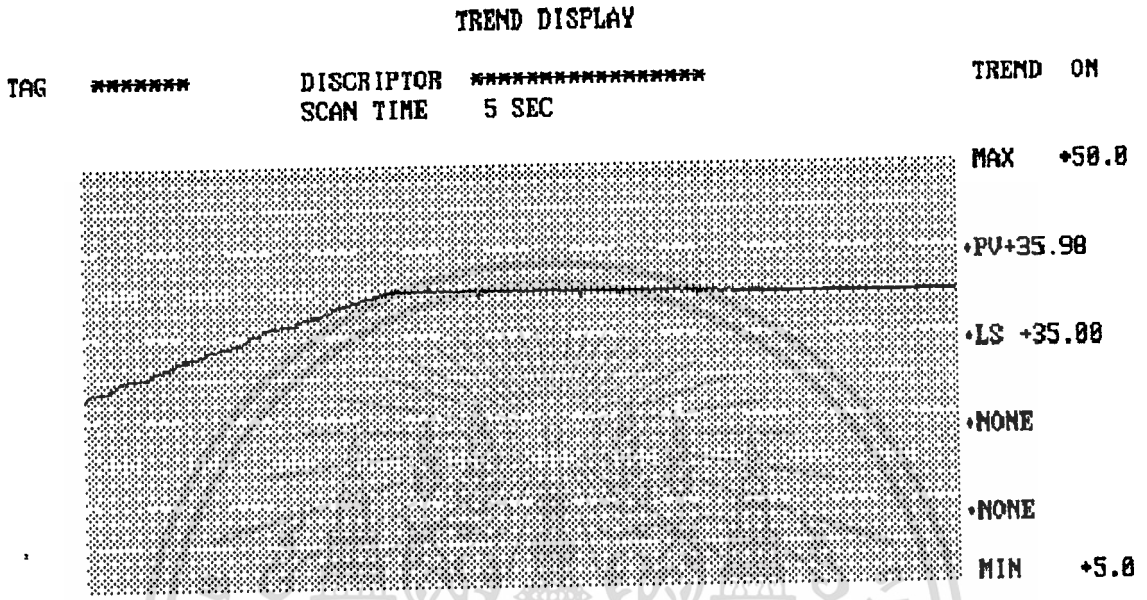
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 4.7 แสดงบล็อคการทำงานของ
โปรแกรม DCS ในการควบคุมอุณหภูมิน้ำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลการทดลอง



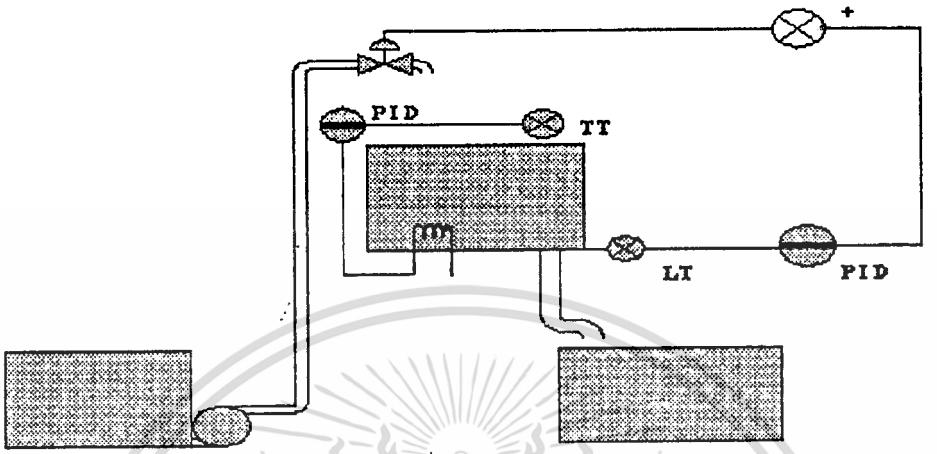
รูป 4.8 รูปแสดงผลการทดลองการควบคุมอุณหภูมิของน้ำ

จากการทดลองได้ค่าพารามิเตอร์ของตัวควบคุมดังนี้ P=50 , I=30 , D=0.3

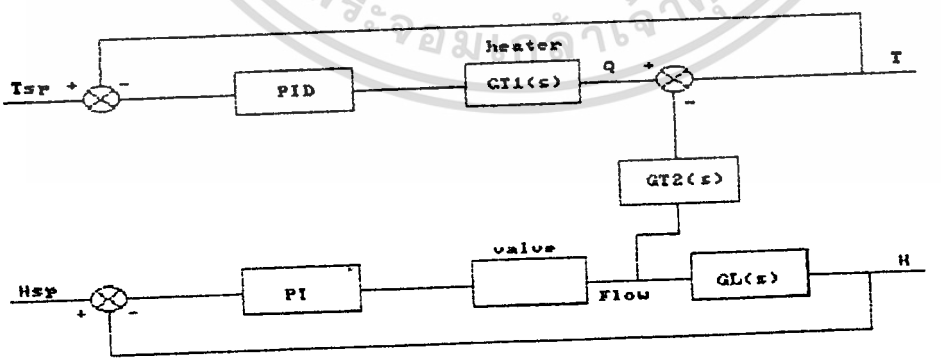
จากผลการทดลองพบว่า การควบคุมอุณหภูมิ อุณหภูมิจะเข้าสู่เซตพอยน์ช้ามาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.การควบคุมระดับและอุณหภูมิของน้ำ 1 (LEVEL AND TEMPERATURE CONTROL 1)



รูปที่ 4.9 การควบคุมระดับและอุณหภูมิของน้ำ 1
(LEVEL AND TEMPERATURE CONTROL 1)



รูป 4.10 บล็อคไดอะแกรมแสดงการควบคุมระดับและอุณหภูมิน้ำ 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นตอนการดำเนินงาน

1. ต่อสายสัญญาณจาก TEMPERATURE TRANSMITTER (TT)มาเข้าที่ channel 0 ของบอร์ดPCLD 780 (B)

2. ต่อสายสัญญาณจาก LEVEL TRANSMITTER มาเข้าที่ (LT) channel 1 ของบอร์ด PCLD 780 (B)

3. กำหนดค่าใน Input Block Configuration ที่จะรับสัญญาณจาก (TT) เลือกใช้แอดเดรส AIB00

   กำหนดค่าอินพุต (IN) เป็น I01

   กำหนดค่าต่ำสุดของเอาต์พุต (MIN) เป็น 0

   กำหนดค่าสูงสุดของเอาต์พุต (MAX) เป็น 100

   กำหนดหน่วย (UNIT) เป็นองศาเซลเซียส (C)

4. กำหนดค่าใน Input Block Configuration ที่จะรับสัญญาณจาก (LT) เลือกใช้แอดเดรส AIB01

   กำหนดค่าอินพุต (IN) เป็น I02

   กำหนดค่าต่ำสุดของเอาต์พุต (MIN) เป็น 5

   กำหนดค่าสูงสุดของเอาต์พุต (MAX) เป็น 24

   กำหนดหน่วย (UNIT) เป็นเซนติเมตร (CM)

5. กำหนดค่าใน Control Block Diagram เลือกใช้แอดเดรส ACB00

   กำหนด TAG เป็น TC

   เลือกฟังก์ชันเป็น PID Control

   กำหนดตัวแปรกระบวนการ PV เป็นค่าแอดเดรสของ IBC คือ AIB00

   กำหนดเซตพอยน์ที่ LS เป็นค่าที่เราต้องการ

   กำหนดค่าต่ำสุดของเอาต์พุต (MIN) เป็น 0

   กำหนดค่าสูงสุดของเอาต์พุต (MAX) เป็น 100

   กำหนดหน่วย (UNIT) เป็นเปอร์เซนต์ (%)

6. กำหนดค่าใน Control Block Diagram เลือกใช้แอดเดรส ACB01

   กำหนด TAG เป็น LC

   เลือกฟังก์ชันเป็น PID Control

   กำหนดตัวแปรกระบวนการ PV เป็นค่าแอดเดรสของ IBC คือ AIB01

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กำหนดเซตพอยน์ที่ LS เป็น ค่าที่เราต้องการ

กำหนดค่าต่ำสุดของเอาต์พุต (MIN) เป็น 0

กำหนดค่าสูงสุดของเอาต์พุต (MAX) เป็น 100

กำหนดหน่วย (UNIT) เป็นเปอร์เซนต์ (%)

7. กำหนดค่าใน Output Block Configuration เลือกใช้แอดเดรส AOB00

กำหนด TAG เป็น TV

กำหนดทิศทางการบังคับวาล์ว (DIRECTION) เป็น DIRECT

กำหนดค่าอินพุต (IN) เป็น TAG ของ CBD คือ TC

กำหนดค่าต่ำสุดของเอาต์พุต (MIN) เป็น 0

กำหนดค่าสูงสุดของเอาต์พุต (MAX) เป็น 100

กำหนดหน่วย (UNIT) เป็นเปอร์เซนต์ของการให้ความร้อน (%)

กำหนดค่าเอาต์พุต (OUT) เป็น Q01

8. กำหนดค่าใน Output Block Configuration เลือกใช้แอดเดรส AOB01

กำหนด TAG เป็น LV

กำหนดทิศทางการบังคับวาล์ว (DIRECTION) เป็น DIRECT

กำหนดค่าอินพุต (IN) เป็น TAG ของ CBD คือ LC

กำหนดค่าต่ำสุดของเอาต์พุต (MIN) เป็น 0

กำหนดค่าสูงสุดของเอาต์พุต (MAX) เป็น 100

กำหนดหน่วย (UNIT) เป็นเปอร์เซนต์ของการเปิดปิดวาล์ว (%)

กำหนดค่าเอาต์พุต (OUT) เป็น Q02

9. ต่อสายสัญญาณจาก channel 0 ของบอร์ดPCLD 780 (A) มาเข้าที่ Watt Transducer (WT) ของกระบวนการ

10. ต่อสายสัญญาณจาก channel 1 ของบอร์ดPCLD 780 (A) มาเข้าที่ control valve (CV) ของกระบวนการ

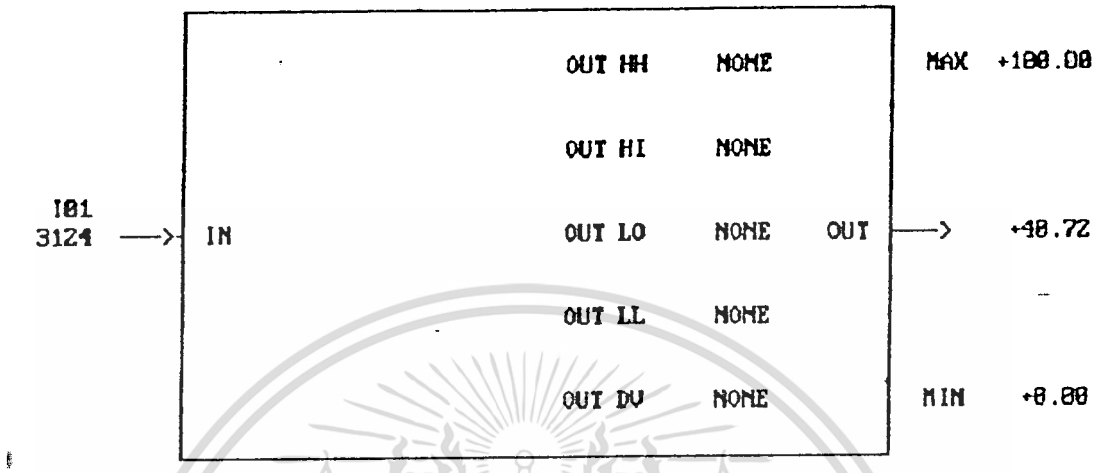
11. กำหนดค่าพารามิเตอร์ของตัวควบคุมที่ Control Block Faceplate

12. กำหนดโหมดการทำงานใน IBC, OBC, CBD เป็นอัตโนมัติ (AUT)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

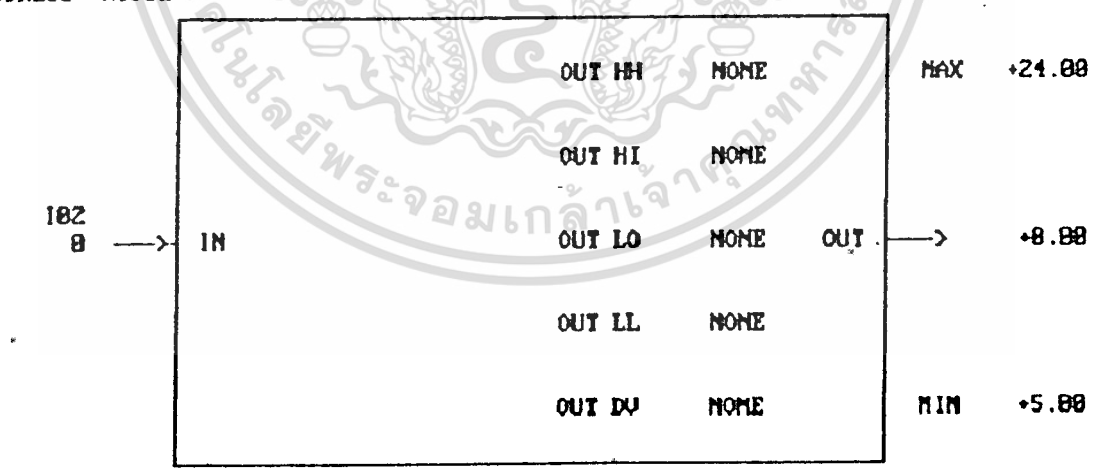
INPUT BLOCK CONFIGURATION

TAG TT DISCRIPTOR TEMP TRANSMITTER MODE AUT
ADDRESS AIB00 UNIT CM

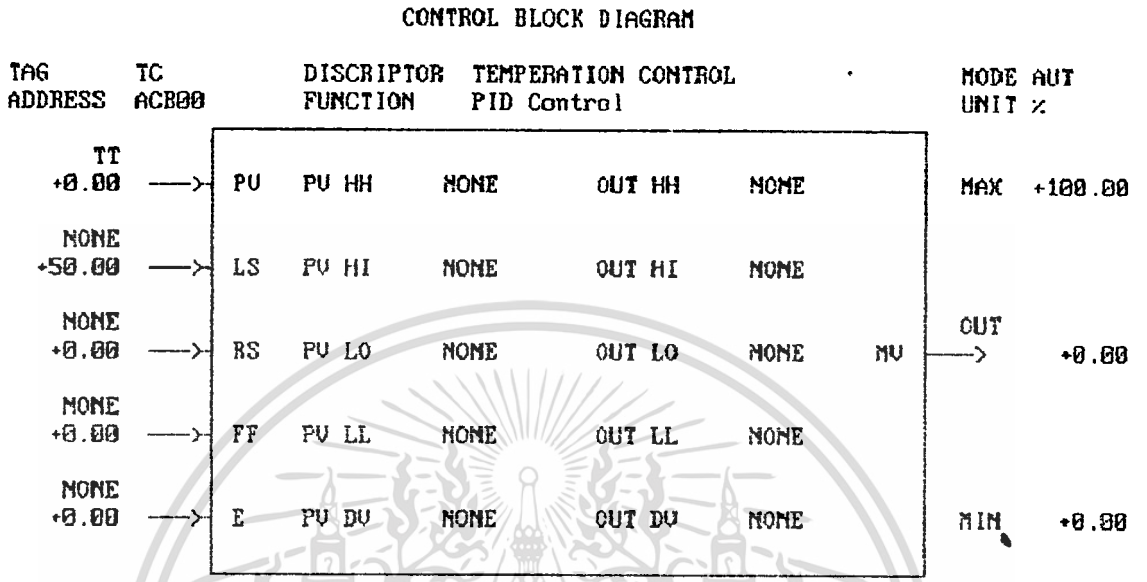


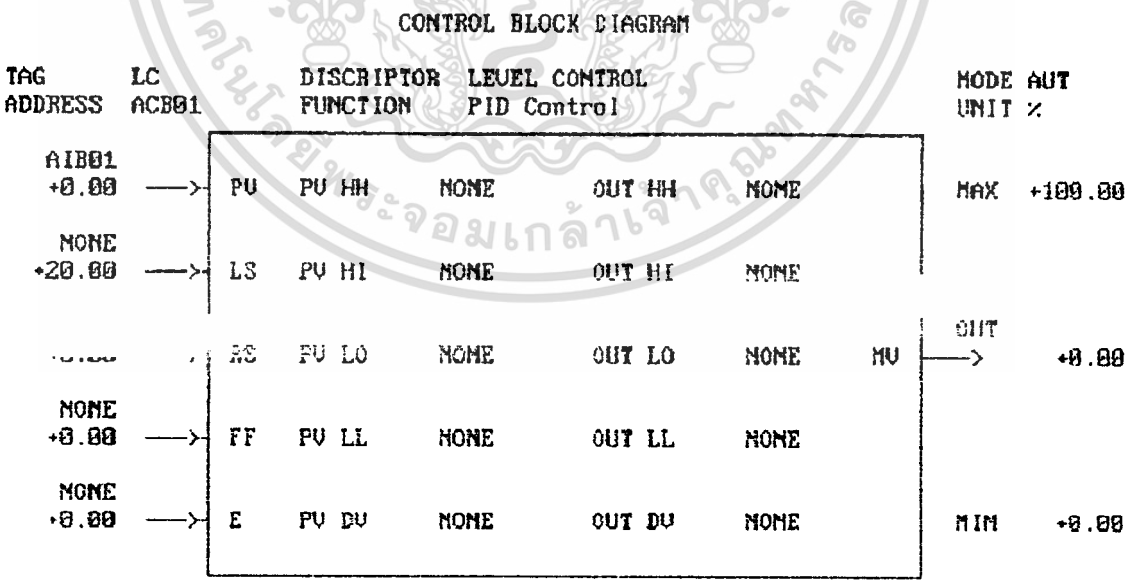
INPUT BLOCK CONFIGURATION

TAG LT DISCRIPTOR LEVEL TRANSMITTER MODE MAN
ADDRESS AIB01 UNIT CM



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้





เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CONTROL BLOCK FACEPLATE

MODE AUT

TAG TC DISCRIPTOR TEMPERATION CONTROL
ADDRESS ACB00 FUNCTION PID Control

PV TT	+0.00	E	NONE	+0.00
LS NONE	+50.00	F	NONE	+0.00
RS NONE	+0.00	G	NONE	+0.00
FF NONE	+0.00	H	NONE	+0.00

Kp Gain	+50.00	
Ti Int Time	+30.00	sec.
Td Dev Time	+0.30	sec.
Kf FF Gain	+0.00	

PV 0.000
LS 50.00
RS 0.000
FF 0.000
E 0.000
F 0.000
G 0.000
H 0.000

OUT = Kp*(SP-PV)+Σ(Ts/Ti)*(SP-PV)+Kp*(Td/Ts)(ΔPV)+Kf*FF

OUT +0.00 %

CONTROL BLOCK FACEPLATE

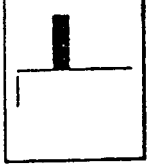
MODE AUT

TAG LC DISCRIPTOR LEVEL CONTROL
ADDRESS ACB01 FUNCTION PID Control

PV TT	+0.00	E	NONE	+0.00
LS NONE	+20.00	F	NONE	+0.00
RS NONE	+0.00	G	NONE	+0.00
FF NONE	+0.00	H	NONE	+0.00

Kp Gain	+100.00	
Ti Int Time	+50.00	sec.
Td Dev Time	+0.00	sec.
Kf FF Gain	+0.00	

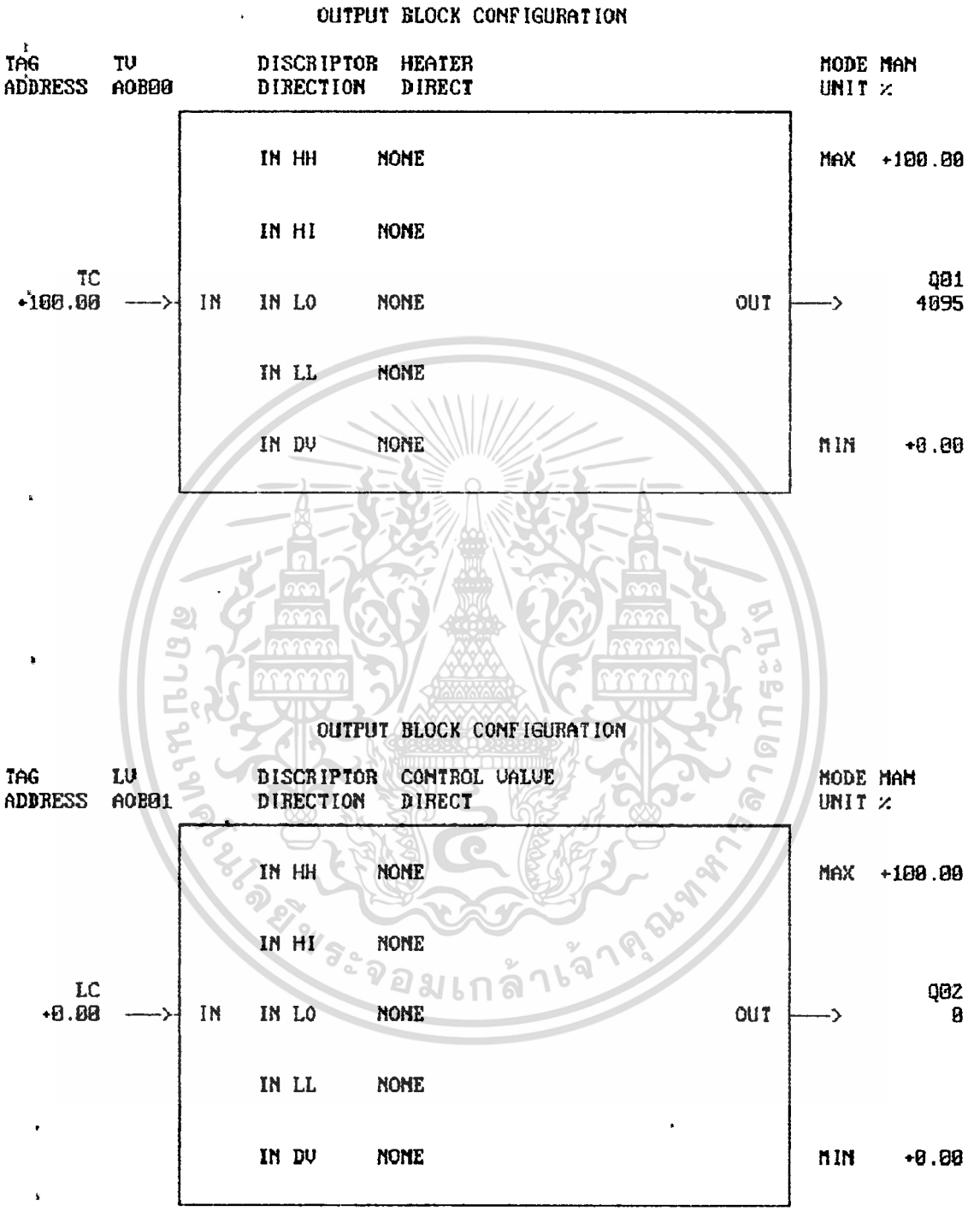
PV 0.000
LS 20.00
RS 0.000
FF 0.000
E 0.000
F 0.000
G 0.000
H 0.000



OUT = Kp*(SP-PV)+Σ(Ts/Ti)*(SP-PV)+Kp*(Td/Ts)(ΔPV)+Kf*FF

OUT +0.00 %

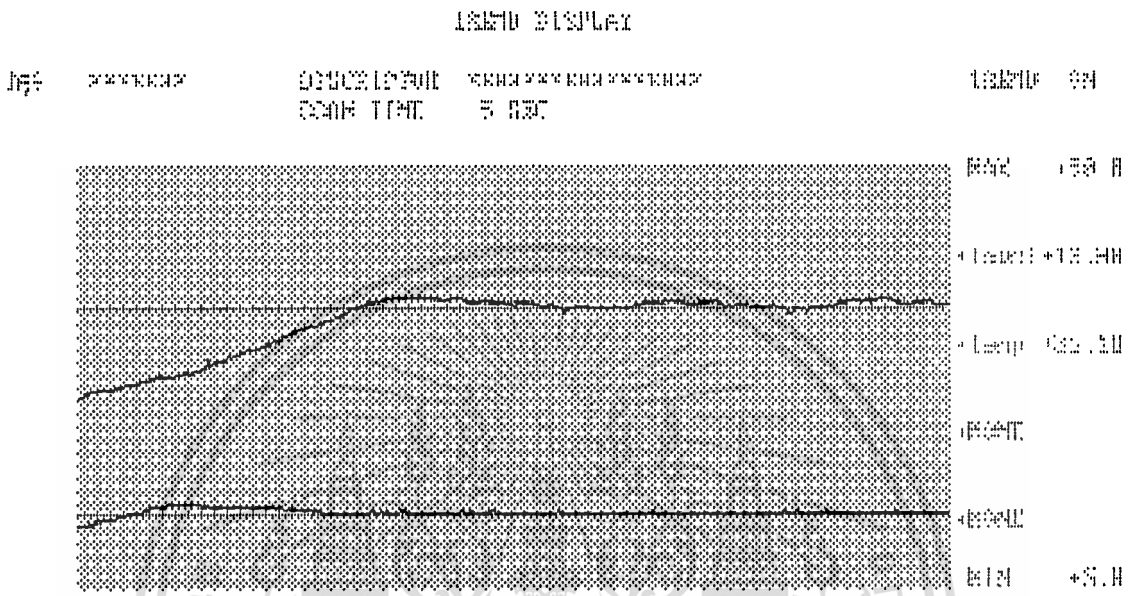
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 4.11 แสดงบล็อคการทำงานของ
โปรแกรม DCS ในการควบคุมอุณหภูมิและระดับน้ำ 1

เอกสารนี้เป็นเอกสารที่ ... ไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลการทดลอง



รูปที่ 4.12 **รูปแสดงผลการทดลองการควบคุมระดับและอุณหภูมิของน้ำ** 1

จากการทดลองได้ค่าพารามิเตอร์ดังนี้

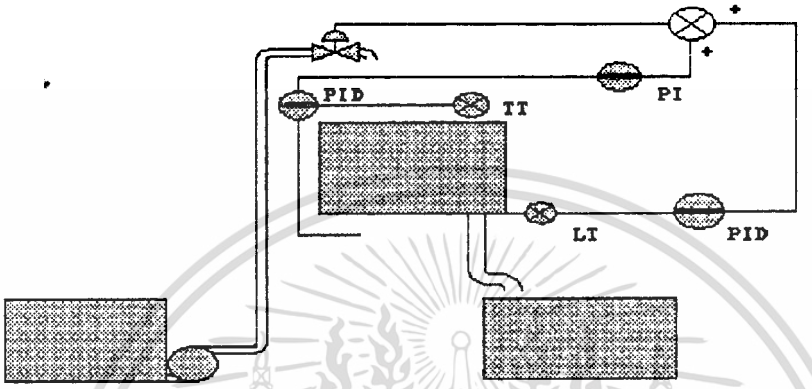
ตัวควบคุม TC ดังนี้ P=50 , I=30 , D=0.3

ตัวควบคุม LC ดังนี้ P=100 , I=50

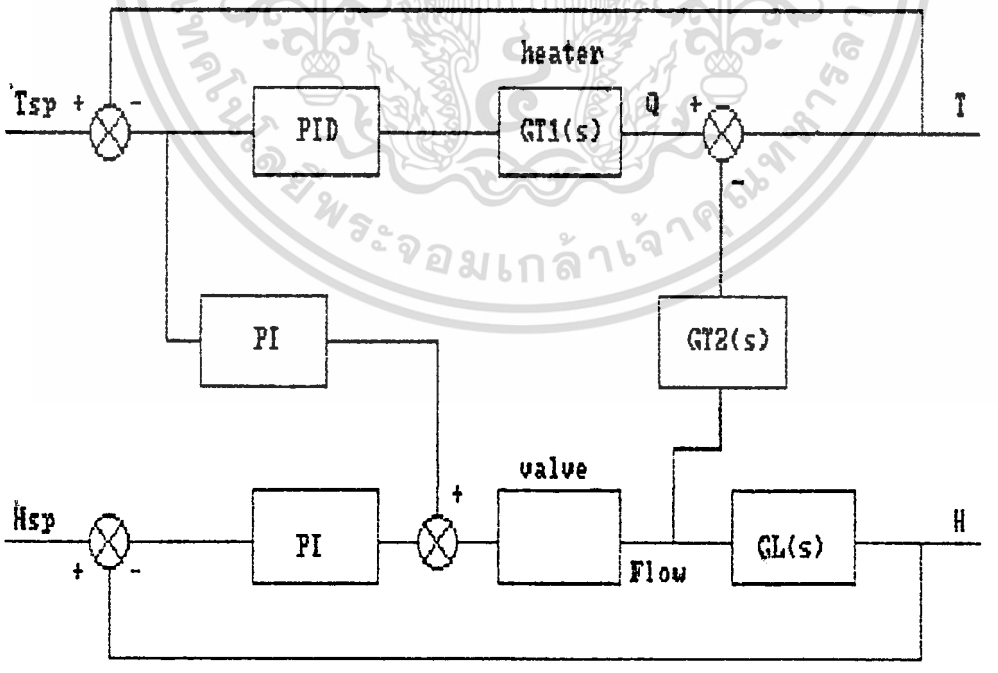
จากผลการทดลองพบว่า การควบคุมระดับน้ำจะเข้าสู่สภาวะที่ได้เสมอ แต่ในการควบคุมอุณหภูมิ อุณหภูมิจะเข้าสู่เซตพอยน์ช้ามาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. การควบคุมระดับและอุณหภูมิของน้ำ 2 (LEVEL AND TEMPERATURE CONTROL 2)



รูปที่ 4.13 **การควบคุมระดับและอุณหภูมิของน้ำ 2**
(LEVEL AND TEMPERATURE CONTROL 2)



รูป 4.14 **บล็อคไดอะแกรมแสดงการควบคุมระดับและอุณหภูมิน้ำ 1**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นตอนการดำเนินงาน

1. ต่อสายสัญญาณจาก TEMPERATURE TRANSMITTER (TT)มาเข้าที่ channel 0 ของบอร์ดPCLD 780 (B)

2. ต่อสายสัญญาณจาก LEVEL TRANSMITTER มาเข้าที่ (LT) channel 1 ของบอร์ด PCLD 780 (B)

3. กำหนดค่าใน Input Block Configuration ที่จะรับสัญญาณจาก (TT) เลือกใช้แอดเดรส AIB00

   กำหนดค่าอินพุต (IN) เป็น I01

   กำหนดค่าต่ำสุดของเอาต์พุต (MIN) เป็น 0

   กำหนดค่าสูงสุดของเอาต์พุต (MAX) เป็น 100

   กำหนดหน่วย (UNIT) เป็นองศาเซลเซียส (C)

4. กำหนดค่าใน Input Block Configuration ที่จะรับสัญญาณจาก (LT) เลือกใช้แอดเดรส AIB01

   กำหนดค่าอินพุต (IN) เป็น I02

   กำหนดค่าต่ำสุดของเอาต์พุต (MIN) เป็น 5

   กำหนดค่าสูงสุดของเอาต์พุต (MAX) เป็น 24

   กำหนดหน่วย (UNIT) เป็นเซนติเมตร (CM)

5. กำหนดค่าใน Control Block Diagram เลือกใช้แอดเดรส ACB00

   กำหนด TAG เป็น TC

   เลือกฟังก์ชันเป็น PID Control

   กำหนดตัวแปรกระบวนการ PV เป็นค่าแอดเดรสของ IBC คือ AIB00

   กำหนดเซตพอยน์ที่ LS เป็น ค่าที่เราต้องการ

   กำหนดค่าต่ำสุดของเอาต์พุต (MIN) เป็น 0

   กำหนดค่าสูงสุดของเอาต์พุต (MAX) เป็น 100

   กำหนดหน่วย (UNIT) เป็นเปอร์เซนต์ (%)

6. กำหนดค่าใน Control Block Diagram เลือกใช้แอดเดรส ACB01

   กำหนด TAG เป็น LC

   เลือกฟังก์ชันเป็น PID Control

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กำหนดตัวแปรกระบวนการ PV เป็นค่าแอดเดรสของ IBC คือ AIB01

กำหนดเซตพอยน์ที่ LS เป็น ค่าที่เราต้องการ

กำหนด FF โดยใส่ชื่อแท็ค ACB02

กำหนดค่าต่ำสุดของเอาต์พุต (MIN) เป็น 0

กำหนดค่าสูงสุดของเอาต์พุต (MAX) เป็น 100

กำหนดหน่วย (UNIT) เป็นเปอร์เซนต์ (%)

7. กำหนดค่าใน Control Block Diagram เลือกใช้แอดเดรส ACB02

กำหนด TAG เป็น CC

เลือกฟังก์ชันเป็น PID Control

กำหนดตัวแปรกระบวนการ PV เท่ากับค่า เซตพอยน์ (LS) ของ TC

กำหนดเซตพอยน์ที่ LS เป็น AIB00

กำหนดค่าต่ำสุดของเอาต์พุต (MIN) เป็น 0

กำหนดค่าสูงสุดของเอาต์พุต (MAX) เป็น 100

กำหนดหน่วย (UNIT) เป็นเปอร์เซนต์ (%)

8. กำหนดค่าใน Output Block Configuration เลือกใช้แอดเดรส AOB00

กำหนด TAG เป็น TV

กำหนดทิศทางการบังคับวาล์ว (DIRECTION) เป็น DIRECT

กำหนดค่าอินพุต (IN) เป็น TAG ของ CBD คือ TC

กำหนดค่าต่ำสุดของเอาต์พุต (MIN) เป็น 0

กำหนดค่าสูงสุดของเอาต์พุต (MAX) เป็น 100

กำหนดหน่วย (UNIT) เป็นเปอร์เซนต์ของการให้ความร้อน (%)

กำหนดค่าเอาต์พุต (OUT) เป็น Q01

9. กำหนดค่าใน Output Block Configuration เลือกใช้แอดเดรส AOB01

กำหนด TAG เป็น LV

กำหนดทิศทางการบังคับวาล์ว (DIRECTION) เป็น DIRECT

กำหนดค่าอินพุต (IN) เป็น TAG ของ CBD คือ LC

กำหนดค่าต่ำสุดของเอาต์พุต (MIN) เป็น 0

กำหนดค่าสูงสุดของเอาต์พุต (MAX) เป็น 100

กำหนดหน่วย (UNIT) เป็นเปอร์เซนต์ของการเปิดปิดวาล์ว (%)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กำหนดค่าเอาต์พุต (OUT) เป็น Q02

10. ต่อสายสัญญาณจาก channel 0 ของบอร์ดPCLD 780 (A) มาเข้าที่ Watt Transducer (WT) ของกระบวนการ

11. ต่อสายสัญญาณจาก channel 1 ของบอร์ดPCLD 780 (A) มาเข้าที่ control valve (CV) ของกระบวนการ

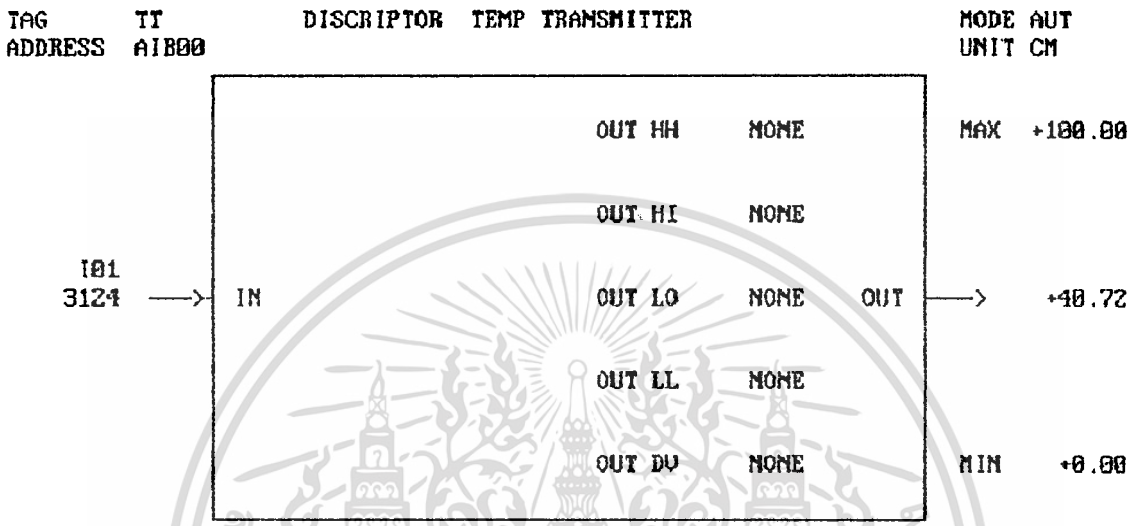
12. กำหนดค่าพารามิเตอร์ของตัวควบคุมที่ Control Block Faceplate

กำหนดค่า Kf เป็น 1

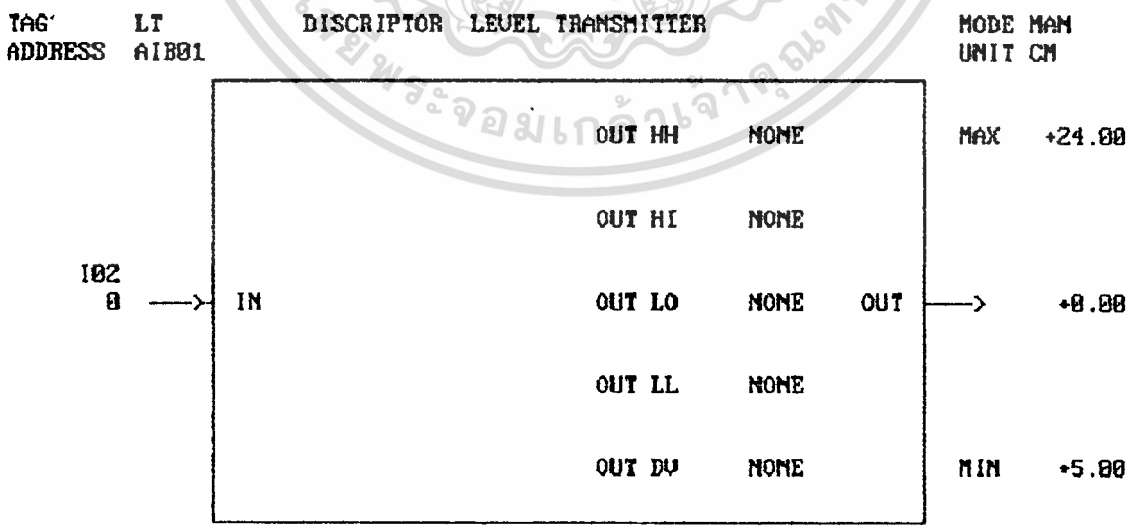
13. กำหนดโหมดการทำงานใน IBC, OBC, CBD เป็นอัตโนมัติ (AUT)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

INPUT BLOCK CONFIGURATION

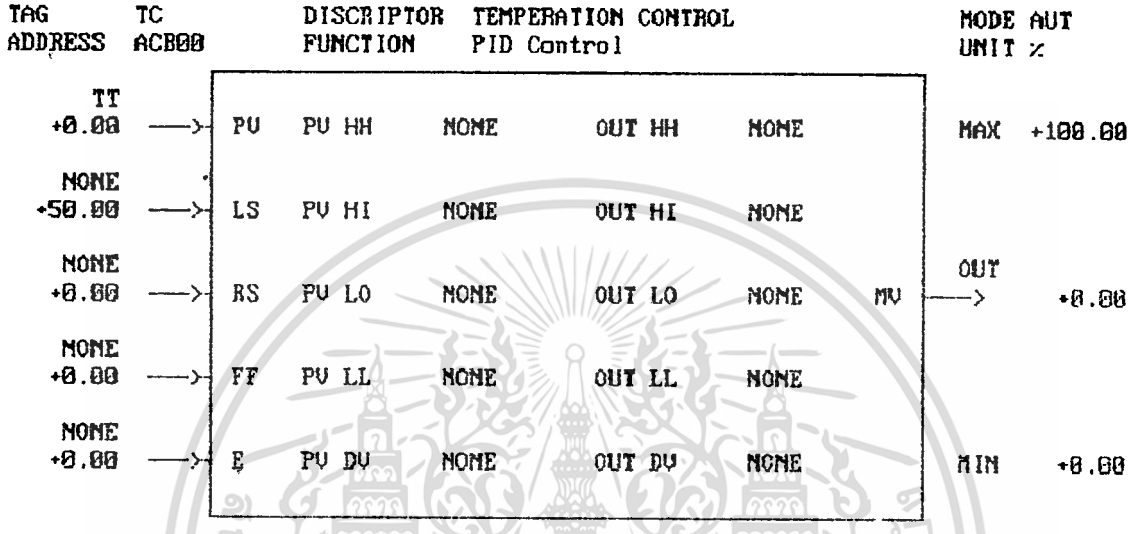


INPUT BLOCK CONFIGURATION

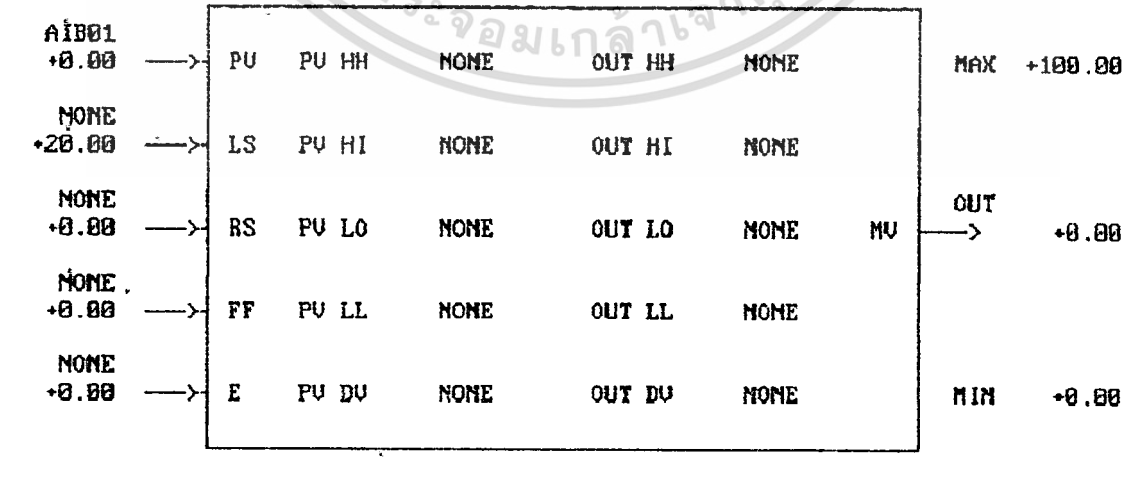


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CONTROL BLOCK DIAGRAM



CONTROL BLOCK DIAGRAM



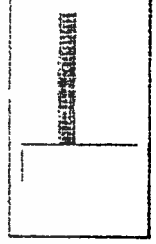
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CONTROL BLOCK FACEPLATE

```
TAG       TC          DISCRIPTOR  TEMPERATION CONTROL
ADDRESS   ACB00       FUNCTION    PID Control

PV TT          +0.00          E  NONE         +0.00
LS NONE       +50.00          F  NONE         +0.00
RS NONE        +0.00          G  NONE         +0.00
FF NONE        +0.00          H  NONE         +0.00

Kp Gain       +50.00
Ti Int Time   +30.00  sec.
Td Dev Time    +0.30  sec.
Kf FF Gain     +0.00
```

MODE AUT

```
PV 0.000
LS 50.00
RS 0.000
FF 0.000
E  0.000
F  0.000
G  0.000
H  0.000
```



OUT = Kp*(SP-PV)+Σ(Ts/Ti)*(SP-PV)+Kp*(Td/Ts)(ΔPV)+Kf*FF

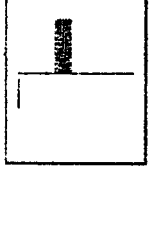
OUT +0.00 %

CONTROL BLOCK FACEPLATE

```
TAG       LC          DISCRIPTOR  LEVEL CONTROL
ADDRESS   ACB01       FUNCTION    PID Control

PV TT          +0.00          E  NONE         +0.00
LS NONE       +20.00          F  NONE         +0.00
RS NONE        +0.00          G  NONE         +0.00
FF CC          +0.00          H  NONE         +0.00

Kp Gain      +100.00
Ti Int Time   +70.00  sec.
Td Dev Time    +0.00  sec.
Kf FF Gain     +0.00
```

MODE AUT

```
PV 0.000
LS 20.00
RS 0.000
FF 0.000
E  0.000
F  0.000
G  0.000
H  0.000
```

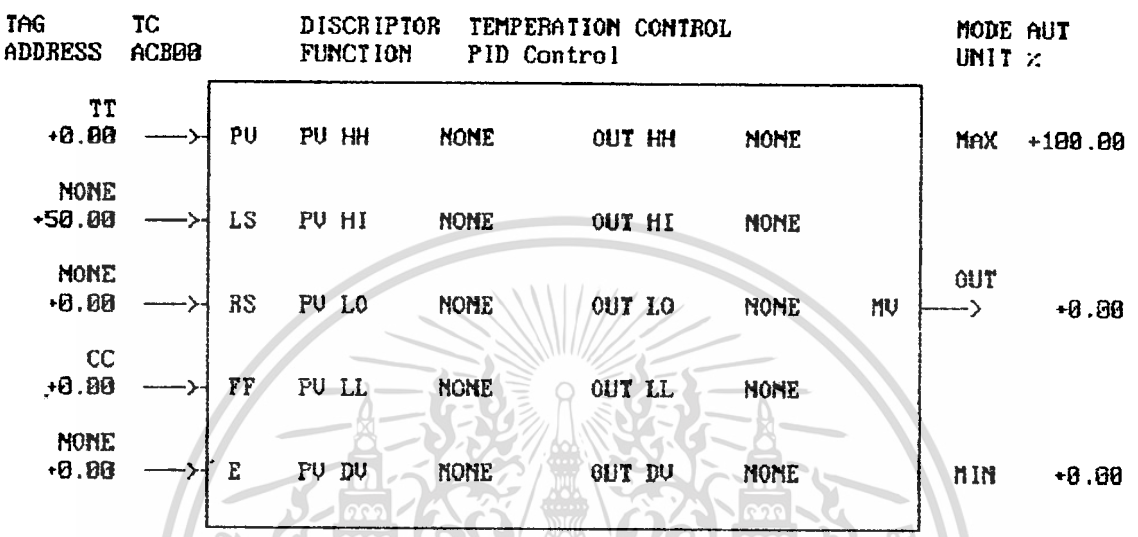


OUT = Kp*(SP-PV)+Σ(Ts/Ti)*(SP-PV)+Kp*(Td/Ts)(ΔPV)+Kf*FF

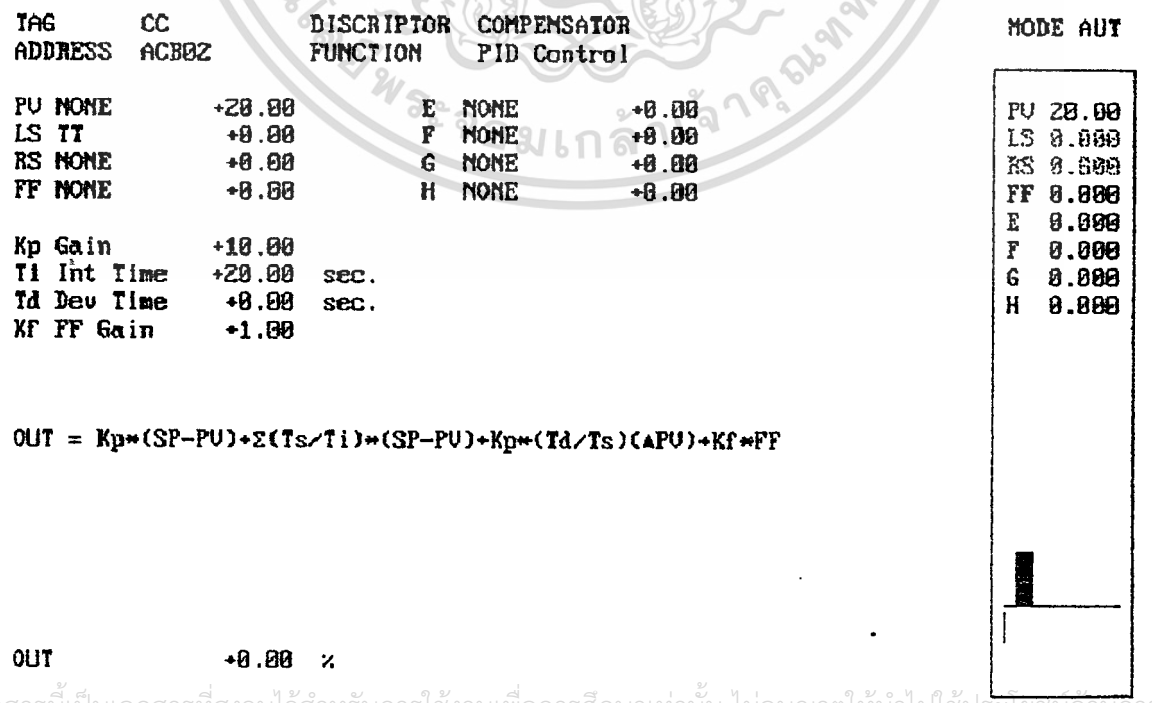
OUT +0.00 %

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

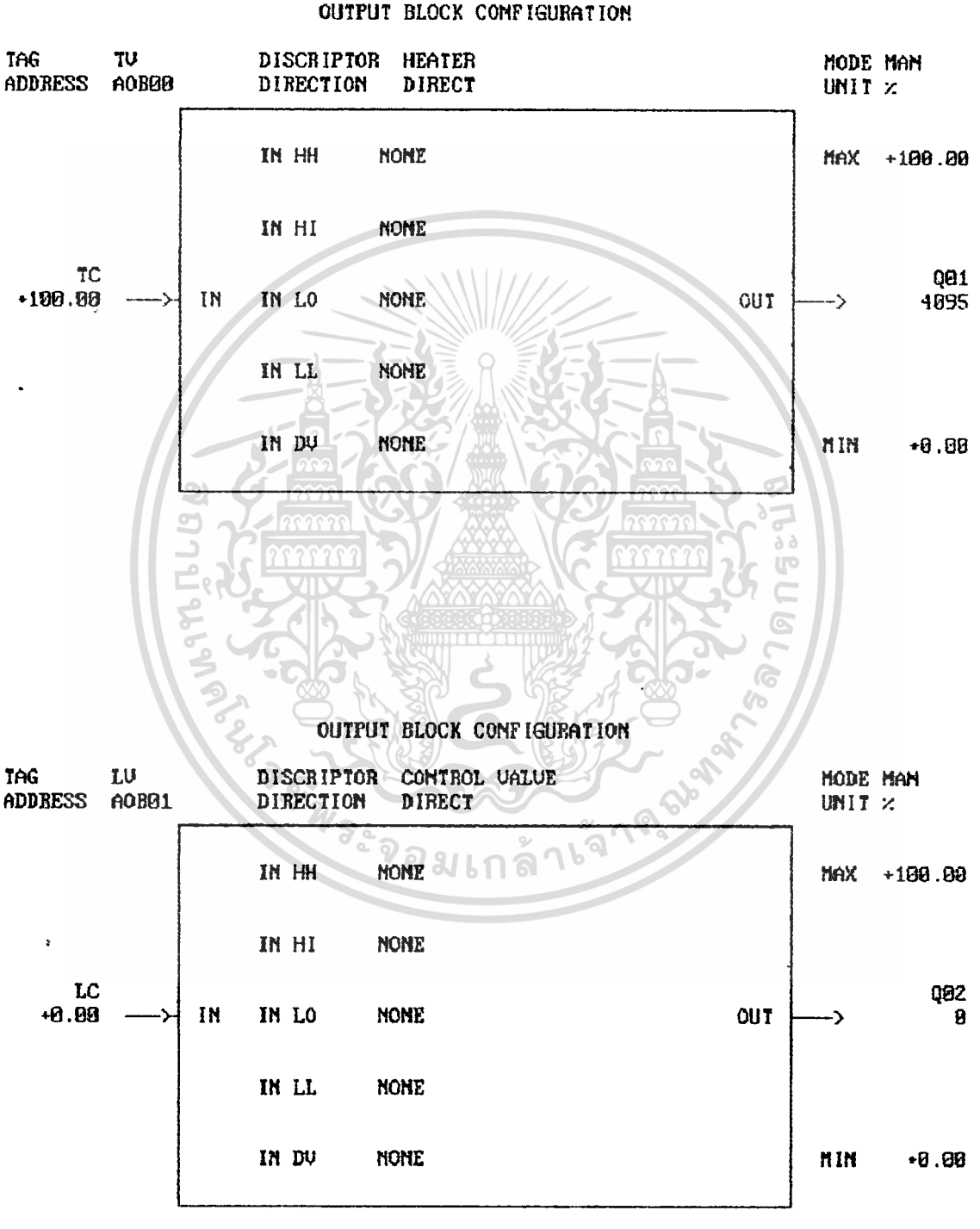
CONTROL BLOCK DIAGRAM



CONTROL BLOCK FACEPLATE



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

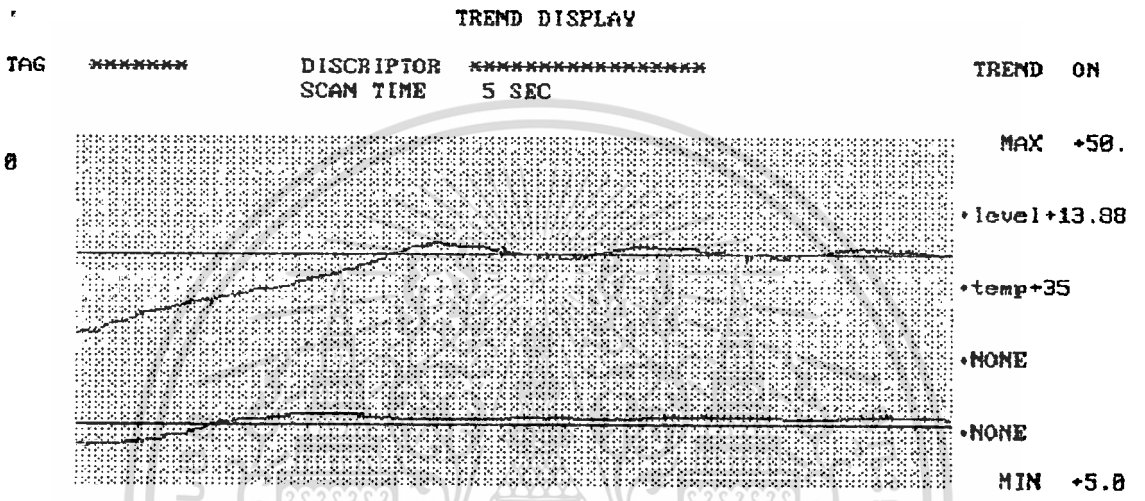


รูป 4.15 **แสดงบล็อคการทำงานของ**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรม DCS ในการควบคุมอุณหภูมิและระดับน้ำ 2

ผลการทดลอง



รูปที่ 4.16 **รูปแสดงผลการทดลองการควบคุมระดับและอุณหภูมิของน้ำ** 2

จากการทดลองได้ค่าพารามิเตอร์ดังนี้

ตัวควบคุม TC ดังนี้ P=50 , I=30 , D=0.3

ตัวควบคุม LC ดังนี้ P=100 , I=70

ตัวควบคุม CC ดังนี้ P=10 , I=20

จากผลการทดลองพบว่า การควบคุมระดับน้ำจะเข้าสู่สภาวะคงที่ได้เสมอ และในการควบคุมอุณหภูมิ อุณหภูมิจะเข้าสู่เซตพอยน์ในเวลาเร็วกว่าเดิม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
สรุปและวิจารณ์

1. การควบคุมระดับน้ำ (LEVEL CONTROL)

จากการทดลองเราพบว่า คอนโทรลวาล์วที่ใช้ในการควบคุมอัตราการไหลของน้ำเข้า จะมีลักษณะเป็นแบบ เปิด/ปิด (ON / OFF) เมื่อวาล์วเปิดน้อยกว่า 15% จะไม่มีอัตราการไหลของน้ำเข้า แต่เมื่อวาล์วเปิดมากกว่า 15% อัตราการไหลของน้ำเข้าจะเท่ากับค่าสูงสุด ซึ่งทำให้การควบคุมอัตราการไหลทำได้ยากมาก แต่เนื่องจากพื้นที่หน้าตัดของถังกระบวนการมีขนาดใหญ่ ทำให้สามารถทำการควบคุมระดับน้ำได้ โดยใช้ PI โหมด โดยให้ค่า P มากๆ และมีค่าผิดพลาด (error) เล็กน้อยซึ่งยอมรับได้

2.การควบคุมอุณหภูมิของน้ำ (TEMPERATURE CONTROL)

ในตอนแรกที่ยังไม่มีการให้น้ำเข้าหรือออกจากถังกระบวนการ โดยกักน้ำไว้ในถังอย่างเดียวแล้วควบคุม เครื่องทำความร้อนอย่างเดียว จะพบว่าระบบไม่สามารถระบายความร้อนออกในช่วงที่อุณหภูมิของน้ำเกินระดับที่ต้องการได้ ทำให้ระบบเข้าสภาวะคงตัว (Steady state) ได้ช้ามาก

3.การควบคุมอุณหภูมิและระดับของน้ำ (TEMPERATURE AND LEVEL CONTROL)

ในตอนแรกได้มีการปรับอัตราการไหลของน้ำเข้าซึ่งมีอุณหภูมิต่ำมาก ควบคุมอุณหภูมิของน้ำในถังร่วมกับเครื่องทำความร้อน จากการทดลองพบว่า วงควบคุมระดับน้ำไม่ได้รับผลกระทบจากวงควบคุมอุณหภูมิ แต่วงควบคุมอุณหภูมิได้รับผลกระทบจากวงควบคุมระดับน้ำ เนื่องจากกระบวนการที่ใช้ในการทดลองนี้ มีอัตราการไหลของน้ำออกช้ามาก ทำให้ระบบสามารถลดพลังงานความร้อนออกได้ช้าด้วย ระบบจึงเข้าสู่สภาวะคงที่ได้ช้า

จากผลการทดลองตอนที่สอง ได้มีการควบคุมความผิดพลาดของอุณหภูมิ ในขณะที่อุณหภูมิสูงเกินความต้องการ ด้วยตัวควบคุมที่ใช้ PI โหมด เพื่อควบคุมอัตราการไหลของน้ำเข้าซึ่งมีอุณหภูมิต่ำมาก ทำให้ระบบสามารถลดเวลาเฉื่อยของวงควบคุมอุณหภูมิลงได้ แต่จะมีข้อเสียคือ วงควบคุมระดับน้ำจะถูกรบกวนตลอดเวลา ทำให้ได้ผลการตอบสนองไม่ดีเท่าที่ควร

สิ่งที่ต้องปรับปรุงแก้ไข

1. ลักษณะการทำงานของวาล์วซึ่งควบคุมอัตราการไหลของเข้า เป็นแบบ เปิด / ปิด ซึ่งมีสาเหตุจาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ขนาดของปั๊มน้ำมีขนาดใหญ่เกินไป
- คุณลักษณะของคอนโทรลวาล์วเป็นแบบ equal percentage

ดังนั้น ทำให้การควบคุมอัตราการไหลทำได้ยาก

2. จากขนาดที่ใหญ่มากของปั๊ม ทำให้อัตราการไหลของน้ำในส่วนข้องอของท่อมีขนาดสูงมาก ซึ่งเมื่อกระทบกับผนังท่อหรือกับน้ำในถังน้ำเย็นเอง ทำให้อุณหภูมิในถังน้ำเย็นลดความเย็นลงมาก

3. อัตราการไหลของน้ำออกจากถังมีขนาดน้อยมากเมื่อเทียบกับอัตราการไหลของน้ำเข้าสู่ถัง เป็นสัดส่วนประมาณ 1ต่อ 3 ซึ่งมีผลกระทบทำให้การควบคุมอุณหภูมิและระดับของน้ำในถังทำได้ยาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

1. Benjamin C Kuo , Automatic Control System ,5th ed., Prentice Hall International Editions , New Jersey ,1987.
2. F G Shinskey , Process Control System , 3rd ed. McGraw-Hill International Editions, New York , 1988.
3. George Stephannopoulos , Chemical Process Control , Prentice Hall International Editions , New Jersey ,1984.
4. Katsuhiko Ogata , Modern Control Engineering , 2nc ed , Prentice Hall International Editions , New Jersey ,1990.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

รายละเอียดของ

การ์ด PCL-812 และการ์ด PCL-726

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Model PCL - 726

Six Channel
D/A Output Card

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

Table of Contents

1. GENERAL INFORMATION 1
1.1. Description 1
1.2. Features 1
1.3. Specifications 2
1.3.1. Analog Output 2
1.3.2. Digital Input 3
1.3.3. Digital Output 4
1.3.4. General Specifications 4

2. INSTALLATION 5
2.1. Initial Inspection 5
2.2. Jumper Settings 5
2.2.1. Base Address Selection 5
2.2.2. Wait State Selection 6
2.2.3. Reference and Mode Selection 7
2.3. Connector Pin Assignment 8
2.4. Hardware Installation 10
2.5. Software Installation 11

3. SIGNAL CONNECTION 13
3.1. Introduction 13
3.2. Voltage Output Connection 13
3.3. 4 to 20mA Current Output Connection 14
3.4. Programmable Attennuator Connection 15
3.5. Digital Signal Connection 16

4. REGISTER STRUCTURE AND FORMAT 17
4.1. Introduction 17
4.2. I/O Port Address Map 17
4.3. D/A Output Registers 18
4.4. Digital I/O Registers 18

5. APPLICATION 20
5.1. D/A Applications 20
5.2. Digital Input and Output 21

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

APPENDIX A. BLOCK DIAGRAM 22

APPENDIX B. CONNECTOR, SWITCH AND VR LOCATIONS . . 23

APPENDIX C. PC I/O PORT ADDRESS MAP 24

APPENDIX D. CALIBRATION 25
D.1. VR Assignment 25
D.2. D/A Calibration 25

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

1. GENERAL INFORMATION

1.1. Description

The PCL-726 provides six analog output channels on a single PC-BUS add-on card. Each channel can be set individually to any of the following ranges: 0 to 5V, 0 to 10V, +/-5V, +/-10V and 4 to 20mA current output. Designed for rugged environment, the PCL-726 is an ideal and economical solution for industrial application requiring multiple analog voltage and/or current output channels.

In addition to the analog output, PCL-726 also provides 16 channels of digital output and 16 channels of input. The D/I and D/O ports are TTL compatible and designed to be fully compatible with the available daughter boards those offer convenient solution to industrial ON/OFF control and sensing application. These daughter boards include PCLD-782 Isolated D/I Board, PCLD-785 Relay Output Board and PCLD-786 SSR & Relay Driver Board.

A utility program disk is supplied with the purpose to minimize user's software effort. The disk contains ready-to-run calibration program, demonstration program as well as program examples.

1.2. Features

* 6 independent D/A output channels.
* 12 bit resolution, double buffered D/A converters.
* Multiple voltage ranges: +/- 10V, +/- 5V, 0 to 10V, 0 to 5V and 4 to 20 mA current loop (sink).
* All D/A outputs will be at 0 volt initial state after RESET or POWER-ON at either bipolar or unipolar mode.
* 16 digital input and 16 digital output channels. They are all TTL compatible.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

* D/I and D/O ports are fully compatible with the valid PC-LabCard daughter boards :

  - PCLD-782 16 channel opto-isolated D/I board.
  - PCLD-785 16 channel relay output board.
  - PCLD-786 8 channel SSR output & 8 channel relay driver board.

* Furnished utility software disk contains :

  - Calibration program.
  - Demonstration program.
  - Program examples.

* DIP switch selectable wait state (0/2/4/6 wait states) insertion to ensure the compatibility with very high speed PC's.

* Screw terminal board (PCLD-780 or PCLD-781) available for D/A output signal wiring.

1.3. Specifications

1.3.1. Analog Output

Channels : 6 channels.

Resolution : 12 Bits. Double buffered.

Output range : 0 to +5V (unipolar)
0 to +10V (unipolar)
+/- 5V (bipolar)
4 to 20mA current loop (sink).
+/- 10V with external DC or AC reference.

Reference voltage :

Internal : -5V (+/- 0.05V)
-10V (+/- 0.05V)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

External : DC or AC, +/- 10V max.

Conversion type : 12 bit monolithic multiplying.

Analog devices : AD7541AKN or equivalent.

Linearity : +/- 1/2 bit.

Accuracy : +/- 0.012% full scale range.

Temperature draft : 2 PPM/deg C full scale range

Settling time : 70 usec max. with OP-07 output amplifiers.
(5 usec max. with OP-37 output amplifiers upon special request.)

Current loop : 4 to 20 mA constant current sink.

Voltage output drive : +/- 5mA max.

Current loop excitation voltage : Minimum +8V, maximum 36V for 4 to 20mA current loop.

Reset (power-on status) : All D/A channels will be at 0 volt output after reset or power-on, either bipolar or unipolar mode.

1.3.2. Digital Input

Channel : 16 bits.

Level : TTL compatible.

Input low : 0.8V max.

Input high : 2.0V min.

Input load : -0.4mA max. at 0.5V.
0.05mA max. at 2.7V.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.3.3. Digital Output

Channel	: 16 bits.
Level	: TTL compatible.
Output low	: 0.5V max. when sink 8mA.
Output high	: 2.4V min. when source 0.05mA.

1.3.4. General Specifications

Power consumption	:	+5V : 500mA typ., 1A max. +12V : 50mA typ., 110mA max. -12V : 14mA typ., 90mA max.
I/O connector	:	20-pin post header for Analog/Digital I/O ports. Adapter available to convert to 37 pin D-type connector.
I/O base address	:	Requires 16 consecutive address locations. Base address definable by the DIP switches for address line A8 - A4. (Factory setting is hex 2C0).
Operating Temp.	:	0 to +50 deg. C.
Storage Temp.	:	-20 to +65 deg. C.
Weight	:	8.6 oz (242.89 gm).

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. INSTALLATION

2.1. Initial Inspection

Inside the shipping container, you should find this operating manual and the PCL-726 card. The PCL-726 was carefully inspected both mechanically and electrically before being shipped. It should be free of marks and scratches and in perfect working order on receipt.

When unpacking, check the unit for signs of shipping damage (damaged box, scratches, dents, etc). If there is damage to the unit or it fails to meet specifications, notify your local sales representative immediately.

Remove the PCL-726 interface card from its protective packaging by grasping the rear metal panel. Keep the anti-vibration packing. Whenever you remove the card from the PC I/O slot, please store the card in the package for protection.

Discharge any static electricity by touching the back of the system unit before you handle this card. You should avoid contact with materials that create static electricity such as plastic, vinyl, and styrofoam. The board should be handled only by the edges to avoid static electric discharge which may damage the integrated circuits on the PCL-726.

2.2. Jumper Settings

The PCL-726 is designed with easy to use in mind. There are one DIP switch and seven jumpers on the PCL-726 card and the function of each switch is discussed in this section. You may want to refer to Appendix B for the physical location of each switch.

2.2.1. Base Address Selection

Switch name: SW1 position 1 to 5

Most PC peripheral devices and interface cards are controlled through the input/output (I/O) ports. These ports are addressed using the I/O

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

port address space. Appendix C provides a PC I/O port address map to help you locate appropriate addresses for different devices.

The I/O port base address for the PCL-726 is selectable via an 8 position DIP switch. The PCL-726 requires 16 consecutive address locations in the I/O space. Valid addresses are from hex 200 to hex 3F0, however you might have used some of these addresses for other devices. Your PCL-726 base address switch setting is set to hex 2C0 in the factory. If you need to adjust it to some other address range, the switch settings for various base addresses are illustrated as below :

I/O Address Range (Hex)		Switch Position 1	2	3	4	5
	A9	A8	A7	A6	A5	A4
	(Fixed)					
200-20F	1	0	0	0	0	0
210-21F	1	0	0	0	0	1
220-22F	1	0	0	0	1	0
.						
.						
2C0-2CF *	1	0	1	1	0	0
.						
.						
300-30F	1	1	0	0	0	0
.						
3F0-3FF	1	1	1	1	1	1

Note :
- ON = 0, OFF = 1
- A4...A9 correspond to PC bus address lines.
- * means factory setting.
- Switch position 6 is not used.

2.2.2. Wait State Selection

Switch name: SW1 position 7 and 8

Some high speed PC's may require that wait states are inserted to achieve stable data transfer. The PCL-726 can be configured with 0, 2, 4, or 6 wait state delays for each transfer of data. The length of the

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

wait state can be selected with the positions 7 and 8 on SW1, as shown below :

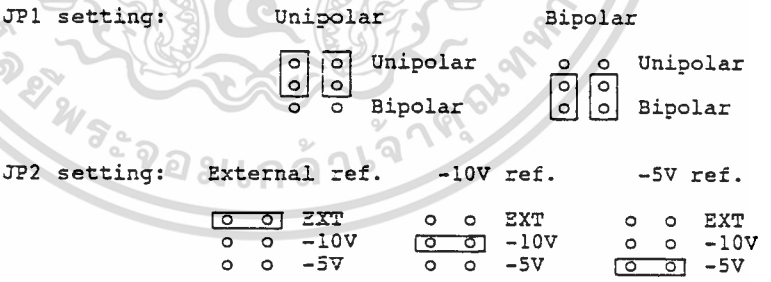
Switch Position 7	8		Wait state time delay
0	0	*	0
1	0		2
0	1		4
1	1		6

Note : * means factory setting

2.2.3. Reference and Mode Selection

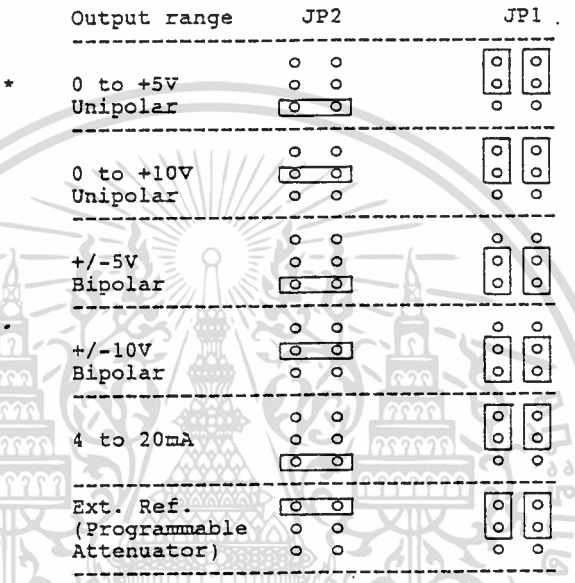
Jumper name : JP1, JP2 of each channel

There are two jumpers for D/A reference source and output mode selections on each D/A conversion channel. The JP1 setting selects the reference of -5V, -10V or external voltage. The JP2 setting selects bipolar or unipolar mode of D/A output.



Some typical combinations of JP1 and JP2 are shown as following:

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า



Note : 1. For programmable attenuator function should, the input signal should be connected to the "ref in" pin of CN3 and CN4.
2. * means factory setting

2.3. Connector Pin Assignment

The PCL-726 is equipped with two 20-pin insulation displacement (mass termination) connectors, accessible from the rear plate, and two other 20-pin insulation displacement connectors on-board. All these connectors can be connected to the same type of flat cables, or connected to 37-pin D-type connectors through our PCLK-1050 industrial wiring kit. Please refer to Appendix B for the location of each connector.

The following diagrams illustrate the pin assignment of each connector.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Legend :

V OUT	-	Analog voltage output
I OUT	-	Analog current output
REF OUT	-	Reference voltage output
REF IN	-	Voltage reference input
D/O	-	Digital output
D/I	-	Digital input
D.GND	-	Digital and power supply ground
A.GND	-	Analog ground
NC	-	No connection

Connector 1 (CN1) - Digital Output

D/O 0	1	2	D/O 1
D/O 2	3	4	D/O 3
D/O 4	5	6	D/O 5
D/O 6	7	8	D/O 7
D/O 8	9	10	D/O 9
D/O 10	11	12	D/O 11
D/O 12	13	14	D/O 13
D/O 14	15	16	D/O 15
D.GND	17	18	D.GND
+5V	19	20	+12V

Connector 2 (CN2) - Digital Input

D/I 0	1	2	D/I 1
D/I 2	3	4	D/I 3
D/I 4	5	6	D/I 5
D/I 6	7	8	D/I 7
D/I 8	9	10	D/I 9
D/I 10	11	12	D/I 11
D/I 12	13	14	D/I 13
D/I 14	15	16	D/I 15
D.GND	17	18	D.GND
+5V	19	20	+12V

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Connector 3 (CN3) - D/A Output (Ch #1 to Ch #3)

-5V REF OUT	1	2	-10V REF OUT
-5V REF OUT	3	4	-10V REF OUT
D/A #1 V OUT	5	6	D/A #1 I OUT
D/A #1 REF IN	7	8	A.GND
D/A #2 V OUT	9	10	D/A #2 I OUT
D/A #2 REF IN	11	12	A.GND
D/A #3 V OUT	13	14	D/A #3 I OUT
D/A #3 REF IN	15	16	A.GND
A.GND	17	18	A.GND
NC	19	20	NC

Connector 4 (CN4) - D/A Output (Ch #4 to Ch #6)

D/A #4 V OUT	1	2	D/A #4 I OUT
D/A #4 REF IN	3	4	A.GND
D/A #5 V OUT	5	6	D/A #5 I OUT
D/A #5 REF IN	7	8	A.GND
D/A #6 V OUT	9	10	D/A #6 I OUT
D/A #6 REF IN	11	12	A.GND
A.GND	13	14	A.GND
D.GND	15	16	D.GND
+5V	17	18	+5V
+12V	19	20	+12V

2.4. Hardware Installation

Installing the card in your computer :

1. Turn the computer off. Turn the power off to any peripheral devices (such as printers and monitors).

2. Disconnect the power code and any other cables from the back of the computer. Turn the system unit so the back of the unit faces you.

3. Remove the system unit cover (refer to your computer users guide if necessary).

4. Locate the expansion slots at the rear of the unit and choose any unused slot.

5. Remove the screw that secures the expansion slot cover to the system unit (save the screw to secure the interface card retaining bracket).

6. Carefully grasp the upper edge of the PCL-726 card. Align the hole in the retaining bracket with the hole on top of the expansion slot, and align the gold striped edge connector with the expansion slot socket. Press the board firmly into the socket.

7. Replace the screw in the expansion slot retaining bracket.

8. Attach necessary accessories, e.g., 20 pin flat cable or connector adapter, to the interface card based on your application requirements.

9. Replace the system unit cover. Connect the cables you removed in step 2. Turn the computer power on.

The hardware installation is now completed, proceed to install the software driver.

2.5. Software Installation

A floppy diskette containing utility software is included with each PCL-726 to minimize your application programming work and support the PCL-726 calibration. The utility programs include :

1. Testing program.

2. Demonstration programs.

3. Calibration program.

It is strongly recommended that you make a working copy from the master copy diskette and save the master copy in safe place. You may use the DOS COPY or DISKCOPY commands to copy the diskette files to another floppy disk or use the COPY command to copy the files to a hard disk. Refer to your MS-DOS users manual for details of how to

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

make backup copies of your diskettes.

Since the operation of the calibration program will be covered in Appendix D, we will only explain the functions of these program files.

* TEST726.EXE : The object of the test program is to test the PCL-726 functions easily by the user. The test functions consist of D/A voltage output, D/A current loop output, digital output and digital input.

* DEMO726A.BAS : This demonstration program provides an example of writing I/O ports to program D/A #1 to generate +3V at unipolar mode and than programs D/A #2 to generate -3V at bipolar mode. Additionally, there are also some explanations about how to program a current output of 10mA on D/A #3

* DEMO726B.BAS : This demonstration program provides an example that programs D/A #1 to D/A #6 to generate a voltage waveform rising from -5V to +5V and then a drop from +5V to -5V at bipolar mode.

* CALB726.BAS : This calibration program provides a procedure to calibrate the voltage and current output of each D/A channels. The program will guide the user to adjust the variable resistors to get the best accuracy of the outputs.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

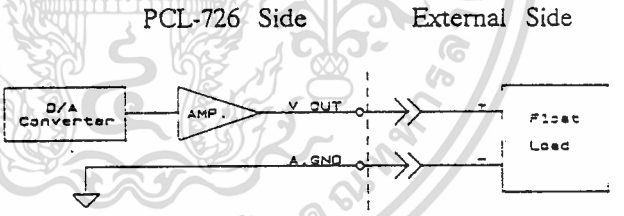
3. SIGNAL CONNECTION

3.1. Introduction

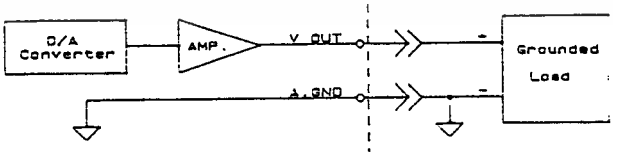
Correct signal connection is one of the most important steps to assure your application system will send and receive data correctly. Since most data acquisition applications involve voltage measurement, correct signal connection will avoid costly damage to your equipment. This section provides some useful information on signal connection in different types of data acquisition applications.

3.2. Voltage Output Connection

The PCL-726 supports 6 channels of D/A voltage output. There is only one output signal wire for each channel. The voltage is referred to the common ground. It is fairly simple to connect a voltage output channel to a floating load. A standard wiring diagram is illustrated below :

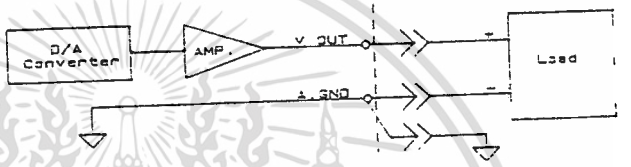


For grounded load, the signal should be connected as:



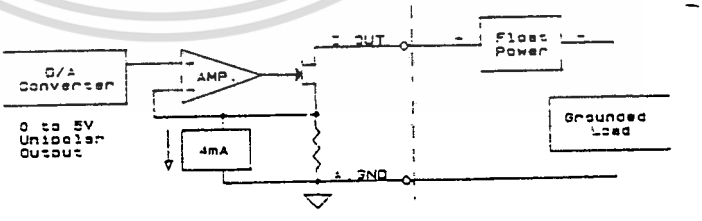
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

For some differential input loads, an external ground wire is needed and the signal connection is recommended as following:

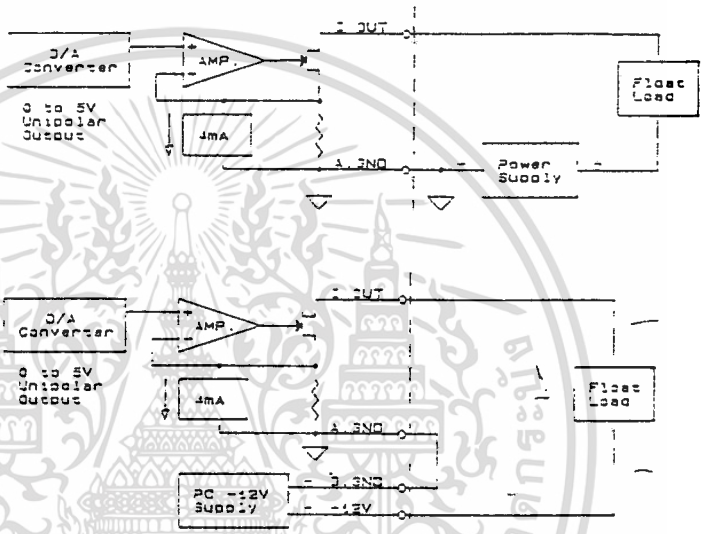


3.3. 4 to 20mA Current Output Connection

The PCL-726 provides 6 channels of 4 to 20mA current output. The current loop output utilizes the 0 to 5V (unipolar) voltage output as the driving source and a current drive circuit consists of a power FET, reverse protection diode and a constant current source. The voltage bias of this current output should be within 8 to 36 volts to insure correct operation. A 24 volt power supply is ideal for this application. The PCL-726 also provides internal 12 volt power source for current loop by user's selection. There are three ways of connecting: a grounded load with a floating supply, a floating load with a grounded power supply, a floating load with internal 12 volt supply. The connections are shown below:

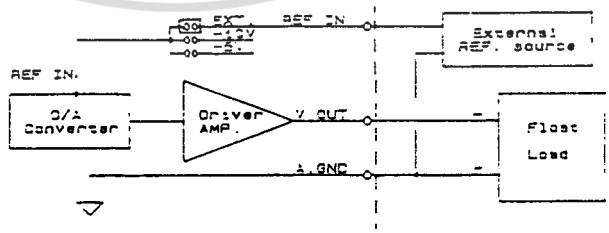


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



3.4. Programmable Attennuator Connection

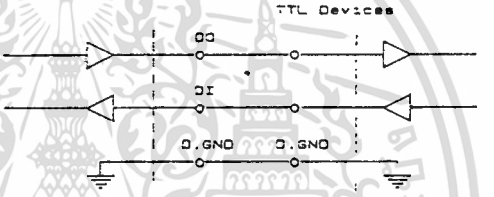
A variety of D/A operations can be supported by your PCL-726. For example, the PCL-726 can function as a digital attenuator by inputing variable AC or DC references, or can be used to generate arbitrary waveform outputs. The connection is :



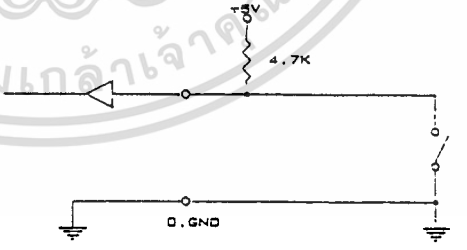
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

3.5. Digital Signal Connection

The PCL-726 has 16 digital input and 16 digital output channels. The digital I/O levels are TTL compatible. To transmit or receive digital signal to/from other TTL devices, the connection is :



To receive an OPEN/SHORT signal from a switch or relay, a pull up resistor must be added to ensure the high level when open.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. REGISTER STRUCTURE AND FORMAT

4.1. Introduction

The PCL-726 requires 16 consecutive addresses in I/O space. The most important issue in programming the PCL-726 is understanding the meaning of the 16 registers addressable from the selected I/O port base address. A summary map of the functions of each address and the data format of each register are given in the following sections.

4.2. I/O Port Address Map

The following table provides the location of each register and driver relative to the base address, and its usage.

Legend :

```
D/A  -   Analog output
D/O  -   Digital output
D/I  -   Digital input
R    -   Read operation on that byte
W    -   Write operation on that byte
```

Address	R/W	Usage
BASE + 0	W	D/A ch #1 high byte data (bit 0-3)
BASE + 1	W	D/A ch #1 low byte data
BASE + 2	W	D/A ch #2 high byte data (bit 0-3)
BASE + 3	W	D/A ch #2 low byte data
BASE + 4	W	D/A ch #3 high byte data (bit 0-3)
BASE + 5	W	D/A ch #3 low byte data
BASE + 6	W	D/A ch #4 high byte data (bit 0-3)
BASE + 7	W	D/A ch #4 low byte data
BASE + 8	W	D/A ch #5 high byte data (bit 0-3)
BASE + 9	W	D/A ch #5 low byte data
BASE + 10	W	D/A ch #6 high byte data (bit 0-3)
BASE + 11	W	D/A ch #6 low byte data
BASE + 12	W	D/O ch 8-15
BASE + 13	W	D/O ch 0-7
BASE + 14	R	D/I ch 8-15
BASE + 15	R	D/I ch 0-7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3. D/A Output Registers

The D/A output registers are write registers using address BASE +0 to base +11.

Data Format :

BASE + 0 (2,4,6,8,10)	D7	D6	D5	D4	D3	D2	D1	D0
D/A # 1 (2,3,4,5,6) High byte	X	X	X	X	DA11	DA10	DA9	DA8
BASE + 1 (3,5,7,9,11)	D7	D6	D5	D4	D3	D2	D1	D0
D/A # 1 (2,3,4,5,6) Low byte	DA7	DA6	DA5	DA4	DA3	DA2	DA1	DA0

LEGEND :

DA11 to DA0 - Digital to analog data. DA0 is the least significant byte (LSB) and DA11 is the most significant byte (MSB) of the D/A data.

X - Don't care

The D/A circuit utilizes a double buffer technique to eliminate the transient stage between the writing operations of high byte and low byte data. The high byte data (only the lower 4 bits are significant) must be written first and it is then latched into a latch for temporary storage and does not change the analog output. While the low byte is being sent, the high byte data reaches the D/A converter at the same time as the low byte data.

NOTE : THE HIGH BYTE DATA MUST BE SENT FIRST.

4.4. Digital I/O Registers

The PCL-726 offers 16 digital input channels and 16 digital output channels. The digital input channels use the I/O address BASE +14 and BASE +15. The digital output channels use the I/O address BASE +12 and BASE +13. The data format of each port is as following :

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Data Format :

BASE + 12	D7	D6	D5	D4	D3	D2	D1	D0
D/O high byte	DO15	DO14	DO13	DO12	DO11	DO10	DO9	DO8
BASE + 13	D7	D6	D5	D4	D3	D2	D1	D0
D/O low byte	DO7	DO6	DO5	DO4	DO3	DO2	DO1	DO0
BASE + 14	D7	D6	D5	D4	D3	D2	D1	D0
D/I high byte	DI15	DI14	DI13	DI12	DI11	DI10	DI9	DI8
BASE + 15	D7	D6	D5	D4	D3	D2	D1	D0
D/I low byte	DI7	DI6	DI5	DI4	DI3	DI2	DI1	DI0

LEGEND :

DO15 to DO0 - Digital output data. DO7 to DO0 is the low byte data and DO15 to DO7 is the high byte data (MSB) of the digital output port.

DI15 to DI0 - Digital input data. DI7 to DI0 is the low byte data and DI15 to DI7 is the high byte data (MSB) of the digital input port.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. APPLICATION

5.1. D/A Applications

The PCL-726 provides 6 D/A channels which use double buffered 12 bit multiplying D/A converters. The D/A registers are write registers using address BASE +0 to BASE +11.

When programming the D/A channels, please note that the most significant byte (high byte data) should be sent first. It is then temporarily held by a register in the D/A and not released to the D/A converter. After the least significant byte (low byte data) is written, the low byte and high byte are added and passed to the D/A converter at the same time. This double buffering process protects the D/A data integrity through a single step update.

The PCL-726 provides an internal precision fixed -5V reference. If this voltage is used as D/A input reference, the D/A output range will be 0 to +5V. You may select other external DC or AC sources as the D/A reference inputs. The maximum reference voltage is +/-10V and the maximum D/A output range is +/- 10V.

Connector CN3 and CN4 support all D/A signal connections. The pin assignment of these connectors is described in Section 2.3. Section 3.2. and 3.3. cover the D/A signal connections and gives an illustration of a wiring diagram.

The PCL-726 D/A functions can be performed in the way by writing I/O instructions to the registers directly. A commonly used D/A application is clearly illustrated in the demonstration program on your diskette. The demonstration program, DEMO726A.BAS, on the PCL-726 software diskette provides a complete explanation of programming either a single D/A channel or more D/A channels.

The following program examples written in BASIC show how to program D/A channel #1 to generate a voltage of +3V.

```
10   ' Figure out the values of the low and high bytes.
20   ' The jumper is set to -5V reference and unipolar
30   ' and the D/A range is 0V to +5V.
```


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการ
กรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
40   V = 3.0
50   X% = (V/5*4096)
60   ' X% = (V/5*2048)+2048 for bipolar -5V to +5V
70   IF X%>4095 OR X%<0 THEN PRINT "OVERFLOW !!!" : END
80   HI% = X% \ 256
90   LO% = X% MOD 256
100  ' Write the value into registers
110  OUT &H2C0+0, HI% : OUT &H2C0+1, LO%
120  END
```

5.2. Digital Input and Output

The PCL-726 provides 16 digital input channels and 16 digital output channels. It is fairiy straight forward to program the digital input and output channels. Attention should be paid to the pin assignments of connectors CN1 (digital output) and CN2 (digital input).

A reading operation on any of the D/I ports will read in the value of the 8 corresponding digital input channels. To access the D/I ports in BASIC, use the following statement:

VALUE = INP (ADDRESS)

Where ADDRESS is BASE+14 (high byte) or BASE+15 (low byte).

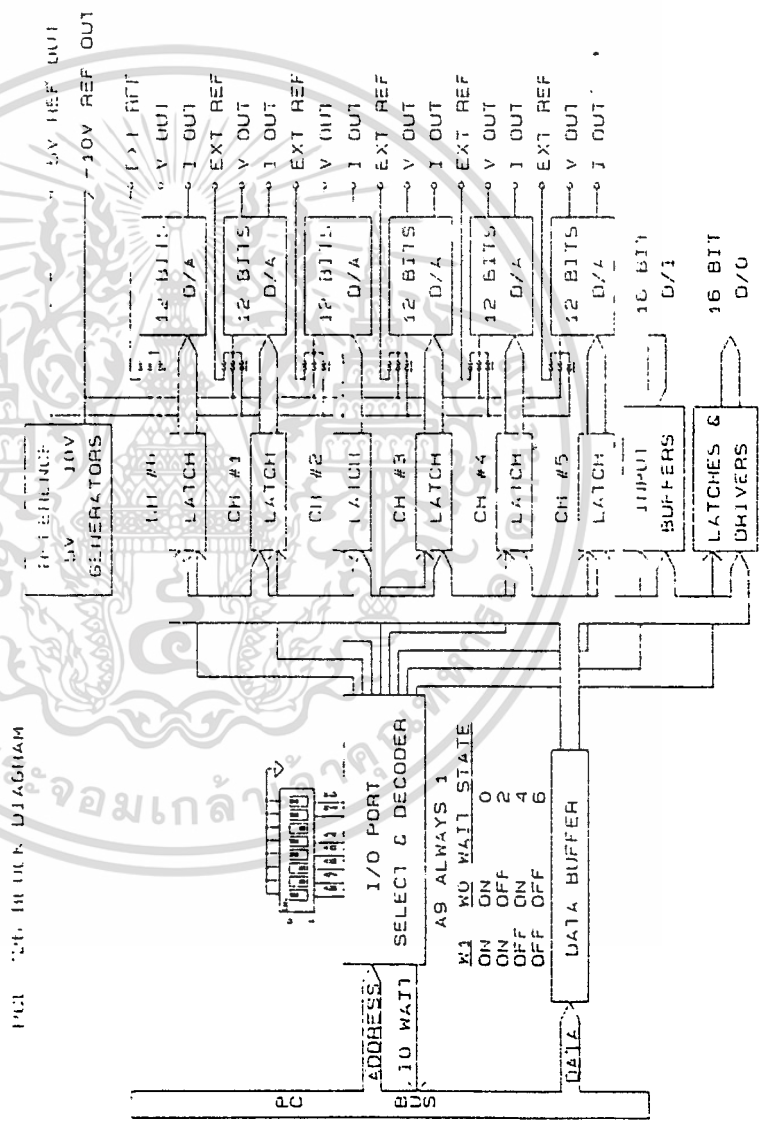
A writing operation to any of the D/O ports will set the desired value of the 8 corresponding digital output channels. To access the D/O ports in BASIC, use the statement shown below:

OUT ADDRESS, VALUE

Where ADDRESS is BASE+12 (high byte) or BASE+13 (low byte).

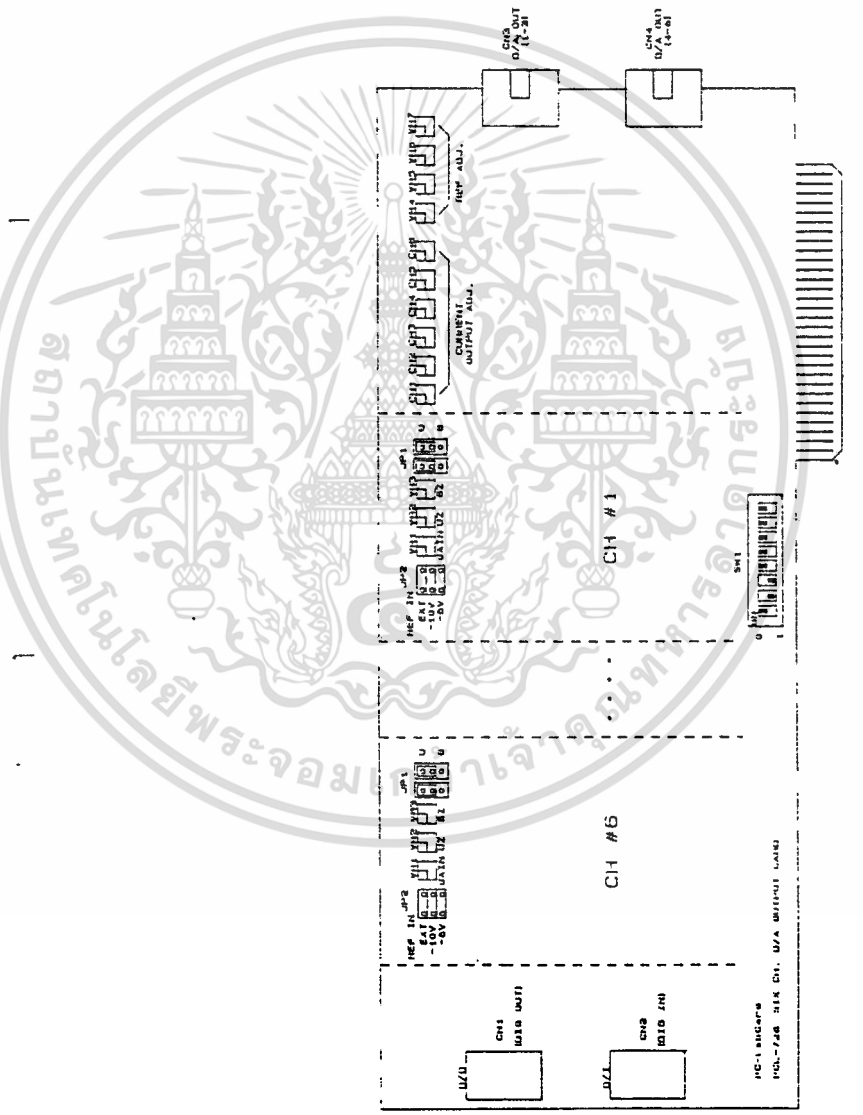
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

APPENDIX A. BLOCK DIAGRAM



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

APPENDIX B. CONNECTOR, SWITCH AND VR LOCATIONS



23

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

APPENDIX C. PC I/O PORT ADDRESS MAP

I/O Address Range (Hex)	Function
000-1FF	Base system
200	Reserved
201	Game control
202-277	Reserved
278-27F	Second printer port
280-2F7	Reserved
2F8-2FF	COM2
300-377	Reserved
378-37F	First printer port
380-3AF	Reserved
3B0-3BF	Mono Disp/Print adapter
3C0-3CF	Reserved
3D0-3DF	Color/Graphics
3E0-3EF	Reserved
3F0-3F7	Floppy disk drive
3F8-3FF	COM1

APPENDIX D. CALIBRATION

In the application of data acquisition and control, it is important to constantly calibrate your measurement device to maintain its accuracy. A calibration program, CALB726.BAS, is provided on the PCL-726 software diskette to assist your calibration work.

D.1. VR Assignment

There are 28 variable resistors (VR) on the PCL-726 to allow you making accurate adjustment on D/A channels. The location of each VR is indicated in Appendix B. Connector, Switch and VR Locations. There are 6 sets of VR1, VR2 and VR3 for the adjustment of 6 D/A channels respectively. The functions of the VR's are listed below:

VR1 : D/A gain adjustment for each channel
VR2 : D/A unipolar offset adjustment for each channel
VR3 : D/A bipolar offset adjustment for each channel
VR4 : -10V reference voltage adjustment for D/A Ch #1 to Ch #3
VR5 : -5V reference voltage adjustment for D/A Ch #1 to Ch #3
VR6 : -10V reference voltage adjustment for D/A Ch #4 to Ch #6
VR7 : -5V reference voltage adjustment for D/A Ch #4 to Ch #6
CH1 : Ch #1 current offset adjustment (4 mA)
CH2 : Ch #2 current offset adjustment (4 mA)
CH3 : Ch #3 current offset adjustment (4 mA)
CH4 : Ch #4 current offset adjustment (4 mA)
CH5 : Ch #5 current offset adjustment (4 mA)
CH6 : Ch #6 current offset adjustment (4 mA)

D.2. D/A Calibration

The calibration program is written in BASIC language. To use it, simply load the program into BASIC and run the program. The default I/O port address setting in the program is hex 2C0. If address hex 2C0 had been occupied by another device, you may either replace that device with the PCL-726 on address hex 2C0 or modify the address setting in the calibration program. It is strongly recommended that do not change

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการ
ค้ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

the program unless you thoroughly understand the PCL-726 calibration program.

Once the calibration program has been loaded and executed, it uses the graphic display and prompts to guide you through the calibration process.

In addition to the calibration program, it is necessary to have a 5 1/2 digit multimeter to perform the calibration. A slot extension card will make your access to the VR's easier. The PC-LabCARD product PCL-755A (Slot Extension Card) is an ideal product to support the slot extension.

A standard procedure to calibrate the PCL-726 card is illustrated below:

Step 1 : Calibrate REFERENCE VOLTAGE SOURCES including -5V REF1 (Ch #1-Ch #3), -10V REF1(Ch #1-Ch #3), -5V REF2 (Ch #4-Ch #6) and -10V REF2 (Ch #4-Ch #6).<VR4-VR7>

Step 2 : Adjust D/A voltage output OFFSET of unipolar mode.
<< adjust VR2 of each channel >>

Step 3 : Adjust D/A voltage output GAIN of unipolar mode.
<< adjust VR1 of each channel >>

Step 4 : Adjust D/A precision 4mA constant current.
<< adjust VR named "CH1" to "CH6" >>

Step 5 : Adjust D/A voltage output OFFSET of bipolar mode.
<< adjust VR3 of each channel >>

Note : The voltage output GAIN of bipolar mode depends on the GAIN of unipolar mode, so it is not necessary to be adjusted. The current gain depends on a precise fixed resistor and does not need adjustment.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PCL-812 ENHANCED MULTI-LAB CARD

USER'S MANUAL

COPYRIGHT NOTICE

This documentation and the software routines contained in the PCL-812 software diskette are copyrighted, 1989, by Advantech Co., Ltd. All rights are reserved. Advantech Co., Ltd. reserves the right to make improvements to the products described in this manual at any time without notice.

No part of this manual may be reproduced, copied, translated or transmitted, in any form or by any means without the prior written permission of Advantech Co., Ltd. Information provided in this manual is intended to be accurate and reliable. However, Advantech Co., Ltd. assumes no responsibility for its use; nor for any infringements of rights of third parties which may result from its use.

ACKNOWLEDGEMENTS

PC-LabCard is a trademark of Advantech Co., Ltd.
IBM and PC are trademarks of International Business Machines Corporation.
MS-DOS is a trade mark of Microsoft Corporation.
BASIC is a trademark of Dartmouth College. Intel is a trademark of Intel Corporation.

Part No. 90812-000A1 Rev. A1
Printed in Taiwan Mar. 1989

Table of Contents

CHAPTER 1 GENERAL INFORMATION 1

1-1 Introduction to The PCL-812 1

1-2 Key Features 1

1-3 Expansion Capabilities 2

1-4 Software Support 2

1-5 Product Specifications 3
1-5-1 Analog Input (A/D Converter) 3
1-5-2 Analog Output (D/A Converter) 3
1-5-3 Digital Input 4
1-5-4 Digital Output 4
1-5-5 Programmable Timer/Counter 4
1-5-6 Interrupt Channel 5
1-5-7 DMA Channel 5
1-5-8 General Specifications 5

CHAPTER 2 INSTALLATION 6

2-1 Initial Inspection 6

2-2 Jumper Settings 6
2-2-1 Base Address Selection 6
2-2-2 Wait State Selection 7
2-2-3 Bipolar Input Range Selection 8
2-2-4 DMA Channel Selection 8
2-2-5 Trigger Source Selection 8
2-2-6 User's Counter Input Clock Selection 8
2-2-7 IRQ Level Selection 9
2-2-8 D/A Reference Source Selection 9

2-3 Connector Pin Assignment 9

2-4 Hardware Installation 11

2-5 Software Installation 12

CHAPTER 3 SIGNAL CONNECTION 15

3-1 Analog Input Connection 15

3-2 Expanding Analog Inputs 15

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3-3 Analog Output Connection 16
3-4 Digital Signal Connection 17
CHAPTER 4 REGISTER STRUCTURE AND FORMAT 18
4-1 I/O Port Address Map 18
4-2 A/D Data Registers 18
4-3 Mux Scan Register 19
4-4 Digital I/O Registers 19
4-5 D/A Output Registers 20
4-6 PCL-812 Control Register 20
4-7 Programmable Interval Timer-Counter Registers 21
CHAPTER 5 A/D CONVERSION 22
5-1 A/D Data Format and Status register 22
5-2 MUX Setting 22
5-3 Trigger Mode 23
5-4 A/D Data Transfer 23
5-5 How to Execute an A/D Conversion 24
CHAPTER 6 D/A CONVERSION 27
6-1 General Information 27
6-2 D/A Applications 28
CHAPTER 7 DIGITAL INPUT AND OUTPUT 29
CHAPTER 8 PROGRAMMABLE INTERVAL TIMER/COUNTER 30
8-1 The Intel 8253 30
8-2 Counter Read/Write and Control Registers 30
8-3 Counter Operating Modes 31
8-3-1 Mode 0 - Interrupt on Terminal Count 31
8-3-2 Mode 1 - Programmable One-Shot 32
8-3-3 Mode 2 - Rate Generator (Divide by N Counter) . 32
8-3-4 Mode 3 - Square Wave Rate Generator 32
8-3-5 Mode 4 - Software Triggered Strobe 33

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8-3-6 Mode 5 - Hardware Triggered Strobe 33

8-4 Counter Operations 33
8-4-1 Read/Write Operations 33
8-4-2 Counter Latch Operation 34

8-5 Counter Applications 34

CHAPTER 9 PROGRAMMING 36

9-1 Introduction to The PCL-812 Programming 36
9-1-1 Programming without The PCL-812 Driver Routines 36
9-1-2 Programming with The PCL-812 Driver Routines . . 36

9-2 PCL-812 Driver Routines 38
9-2-1 FUNCTION 0 - Initialization 39
9-2-2 FUNCTION 1 - Set MUX Scan Range 40
9-2-3 FUNCTION 3 - Perform Single A/D Conversion . . . 41
9-2-4 FUNCTION 4 - Perform N A/D Conversions and Save to Array Variable 42
9-2-5 FUNCTION 5 - Perform N A/D Conversions and Save to Memory with Interrupt Transfer . 43
9-2-6 FUNCTION 6 - Perform N A/D Conversions and Save to Memory with DMA Transfer . . . 45
9-2-7 FUNCTION 7 - Disable Interrupt and DMA Activity 47
9-2-8 FUNCTION 8 - Check Interrupt and DMA Status . . 47
9-2-9 FUNCTION 9 - Convert Data from Memory to Array Variable 48
9-2-10 FUNCTION 10 - Set-up Counter 0 Operation Mode . 50
9-2-11 FUNCTION 11 - Write Counter 0 Data 50
9-2-12 FUNCTION 12 - Read Counter 0 Data 51
9-2-13 FUNCTION 15 - Set Output on Single D/A Channel 52
9-2-14 FUNCTION 16 - Set Output on Both D/A Channels . 52
9-2-15 FUNCTION 17 - Set Pacer Trigger Rate 53
9-2-16 FUNCTION 21 - Write Digital Output DO0 - DO15 . 54
9-2-17 FUNCTION 22 - Read Digital Input DO0 - DO15 . . 55

9-3 Driver Return Code Listing 55

9-4 Programming Examples 57

CHAPTER 10 DIRECT MEMORY ACCESS (DMA) OPERATION 58

10-1 Introduction to The PC 8237 DMA Controller 58

10-2 How to Use DMA Transfer with The PCL-812 58

APPENDIX A PCL-812 BLOCK DIAGRAM 60

APPENDIX B PCL-812 CONNECTOR, SWITCH AND VR LOCATIONS . . 61

APPENDIX C PC I/O PORT ADDRESS MAP 62

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

APPENDIX D CALIBRATION 63
D.1 PCL-812 VR Assignment 63
D.2 A/D Calibration 64
D.3 D/A Calibration 64

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CHAPTER 1 GENERAL INFORMATION

1-1 Introduction to The PCL-812

The PCL-812 is a high performance, high speed, multi-function data acquisition card for IBM PC/XT/AT and compatible computers. The high-end specifications of this full-sized card, and complete software support from third-party vendors make it ideal for a wide range of applications in industrial and laboratory environments. These applications include data acquisition, process control, automatic testing and factory automation.

1-2 Key Features

* 16 single-ended analog input channels.
* An industrial standard 12-bit successive approximation converter (HADC574Z) to convert analog inputs. The maximum A/D sampling rate is 30 KHz in DMA mode.
* Switch selectable versatile analog input ranges.
  Bipolar: +/- 1V, +/- 2V, +/- 5V, +/- 10V.
* Three A/D trigger modes : Software trigger
  Programmable pacer trigger
  External trigger pulse trigger.
* The ability to transfer A/D converted data by program control, interrupt handler routine or DMA transfer.
* An INTEL 8253-5 Programmable Timer/Counter provides pacer output (trigger pulse) at the rate of 0.5 MHz to 35 minutes-/pulse to the A/D. The timer time base is 2 MHz. One 16-bit counter channel is reserved for user configurable applications.
* Two 12 bit monolithic multiplying D/A output channels. An output range of from 0 to +5 can be created by using the on-board -5V reference. This precision reference is derived from the A/D converter reference. External AC or DC references can also be used to generate other D/A output ranges.
* 16 TTL/DTL compatible digital input, and 16 digital output channels.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

1-3 Expansion Capabilities

To complement the powerful features of the PCL-812, the versatility of the card can be further enhanced with the use of the optional daughter boards listed below:

PCL-789

Amplifier/Multiplexer board. This powerful front end analog input signal conditioning card can multiplex 16 differential inputs to one A/D input channel. A high grade instrumentation amplifier provides switch selectable gains of 0.5, 1, 2, 10, 50, 100, 200, 1000 or any user definable gain.

PCLD-786

AC/DC Power SSR and Relay Driver board. This board provides 8 channels of photo-couple isolated solid state relay modules, plus an additional 8 external relay driver outputs.

PCLD-785

Relay Output board. This board provides 16 SPDT relays driven by the 16-bit digital output channels of the PCL-812.

PCLD-782

Isolated D/I board. This 16-channel opto-isolated digital input board provides an easy way to input digital signals to the PCL-812.

PCLD-780

Wiring Terminal board for easy analog and/or digital I/O connections.

1-4 Software Support

The PCL-812 also provides powerful and easy to use software driver routines which can be accessed by the BASIC CALL statement. With these driver routines, application programming becomes much easier especially when you want to use some of the sophisticated features available with the PCL-812 e.g. interrupt or DMA data transfer.

To provide PCL-812 users with more application support, a variety of excellent third-party application software packages have been integrated with the PCL-812. Currently, the PCL-812 is supported by the following software:

* PC-LabDAS - Turn-key, menu driven, general purpose data

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

acquisition package from Advantech.

* UnkelScope - Menu driven data acquisition software with waveform analysis from Unkel Software Inc.

A complete software product catalog is available free from your local PC-LabCARD representative.

This manual is organized to help you understand the functions of the PCL-812, and serve as a programming reference book.

1-5 Product Specifications

1-5-1 Analog Input (A/D Converter)

Channels :	16 Single-ended.
Resolution :	12 bits.
Input Range :	Bipolar : +/- 10V, +/- 5V, +/- 2V, +/- 1V. All input ranges are switch selectable.
Overvoltage :	Continuous +/- 30V Max.
Conversion type :	Successive Approximation.
Converter :	HADC574Z.
Conversion speed :	30 KHz max.
Accuracy :	0.015 % of reading +/- 1 bit.
Linearity :	+/- 1 bit.
Trigger mode :	Software trigger, on-board programmable timer or external trigger.
Data transfer :	Program control, Interrupt control or DMA.
External trigger :	(TTL compatible, load 0.4mA max. at 0.5V (low) or 0.05mA max. at 2.7V (high)).

1-5-2 Analog Output (D/A Converter)

Channels :	2 channels.
Resolution :	12 Bits.
Output Range :	0 to +5V with fixed -5V reference. +/- 10V

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

with external DC or AC reference.

Reference voltage.:	Internal : -5V (+/- 0.05V). External : DC or AC, +/- 10V max.
Conversion type :	12 bit monolithic multiplying.
Analog devices :	AD754/AKN or equivalent.
Linearity :	+/- 1/2 bit.
Output drive :	+/- 5mA max.
Settling time :	30 microseconds.

1-5-3 Digital Input

Channel :	16 bits.
Level :	TTL compatible.
Input voltage :	Low - 0.8V max. High - 2.0V min
Input load :	Low - 0.4mA max. at 0.5V. High - 0.05mA max. at 2.7V.

1-5-4 Digital Output

Channel :	16 bits.
Level :	TTL compatible.
Output voltage :	Low - Sink 8mA at 0.5V max. High - Source -0.4mA at 2.4V min.

1-5-5 Programmable Timer/Counter

Device :	INTEL 8253-5.
Counters :	3 channels, 16 bit. 2 channels permanently connected to 2 MHz clock as programmable pacer, 1 channel free for user application.
Input, gate :	TTL/DTL/CMOS compatible.
Time base :	2MHz.
Pacer Output :	35 minutes/pulse to 0.015 MHz.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

1-5-6 Interrupt Channel

Level :	IRQ 2 to 7, jumper selectable.
Enable :	Via S0, S1 and S2 of CONTROL register.

1-5-7 DMA Channel

Level :	1 or 3, jumper selectable.
Enable :	Via S0, S1 and S2 of CONTROL register.

1-5-8 General Specifications

Power consumption :	+5V : typ. 500mA, max. 1A +12V : typ. 50mA, max. 100mA. -12V : typ. 14mA, max. 20mA.
I/O connector :	20-pin post header for Analog/Digital I/O ports. Adapter available to convert to 37 pin D-type connector.
I/O base address :	Requires 16 consecutive address locations. Base address definable by the DIP switches for address line A8 - A4. (Factory setting is &H220).
Operating Temp :	0 to +50 deg.C.
Storage Temp :	-20 to +65 deg.C.
Weight :	8.6 oz (242.89 gm).

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CHAPTER 2 INSTALLATION

2-1 Initial Inspection

Inside the shipping container, you should find this operating manual and the PCL-812 card. The PCL-812 was carefully inspected both mechanically and electrically before being shipped. It should be free of marks and scratches and in perfect working order on receipt.

When unpacking, check the unit for signs of shipping damage (damaged box, scratches, dents, etc). If there is damage to the unit or it fails to meet specifications, notify our service department or your local sales representative immediately. Also, call the carrier immediately and retain the shipping carton and packing material for inspection by the carrier. We will then make arrangements to repair or replace the unit.

Remove the PCL-812 interface card from its protective packaging by grasping the rear metal panel. Keep the anti-vibration packing. Whenever you remove the card from the PC expansion slot, please store the card in the package for protection.

Discharge any static electricity by touching the back of the system unit before you handle the board. You should avoid contact with materials that create static electricity such as plastic, vinyl, and styrofoam. The board should be handled only by the edges to avoid static electric discharge which could damage the integrated circuits on the PCL-812.

2-2 Jumper Settings

The PCL-812 was designed with easy to use in mind. There are two dip switches and seven jumpers on the PCL-812 interface control card and the function of each switch is discussed in this section. You may want to refer to Appendix B for the physical location of each switch.

2-2-1 Base Address Selection

Switch name: SW1

Most PC peripheral devices and interface cards are controlled through the input/output (I/O) ports. These ports are addressed using the I/O port address space. Appendix C provides a PC I/O port address map to help you locate appropriate addresses for different devices.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

The I/O port base address for the PCL-812 is selectable via an 8 position dip switch. The PCL-812 requires 16 consecutive address locations in I/O space. Valid addresses are from Hex 200 to Hex 3F0, however you might have used some of these addresses for other devices. Your PCL-812 base address switch setting is set to Hex 220 in the factory. If you need to adjust it to some other address range, the switch settings for various base addresses are illustrated as below:

I/O Address Range (Hex)	Switch Position 1	2	3	4	5	6
	A9	A8	A7	A7	A5	A4	A3
	(Fixed)						
200-20XF	1	0	0	0	0	0	X
210-21XF	1	0	0	0	0	1	X
220-22XF*	1	0	0	0	1	0	X
220-23XF	1	0	0	0	1	1	X
300-30XF	1	1	0	0	0	0	X
3F0-3FXF	1	1	1	1	1	1	X

Note : - ON = 0, OFF = 1
- A4...A9 correspond to PC bus address lines.
- * means factory setting.

2-2-2 Wait State Selection

Some high speed PC's may require that a wait state is added to the PCL-812 to achieve stable data transfer. The PCL-812 can be configured with 0, 2, 4, or 6 wait state delays for each transfer of data. The length of the wait state can be selected with the pins 7 and 8 on SW1, as shown below:

Switch Position 7	8	Time Delay
0	0	0
1	0	2
0	1	4
1	1	6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

2-2-3 Bipolar Input Range Selection

Switch name: SW2

The specific analog input range within the bipolar group is selected by this 5 position DIP switch. The following table illustrates the switch setting and corresponding input ranges.

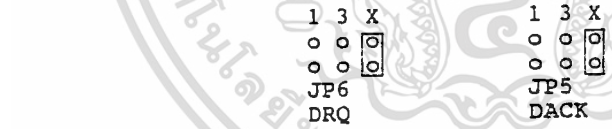
Switch Position 1	2	3	4	5	Bipolar Range
ON	OFF	ON	OFF	X	+/- 10V
OFF	ON	ON	OFF	X	+/- 5V
ON	OFF	OFF	ON	X	+/- 2V
OFF	ON	OFF	ON	X	+/- 1V

Note : See Chapter 5-2 for the user definable analog input range.

2-2-4 DMA Channel Selection

Jumper name: JP5, JP6

The PCL-812 provides DMA data transfer capability. The selection of DMA level 1 or level 3 is controlled by this jumper.



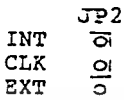
2-2-5 Trigger Source Selection

Jumper name : JP1

2-2-6 User's Counter Input Clock Selection

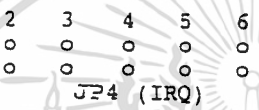
Jumper name : JP2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



2-2-7 IRQ Level Selection

Jumper name : JP4



2-2-8 D/A Reference Source Selection

Jumper name : JP7, JP8

2-3 Connector Pin Assignment

The PCL-812 is equipped with two 20-pin insulation displacement (mass termination) connectors, accessible from the rear plate, and three other 20-pin insulation displacement connectors on-board. All these connectors can be connected to the same type of flat cables, or connected to 37-pin D-type connectors through our PCLK-1050 industrial wiring kit. Please refer to Appendix B for the location of each connector.

The following diagrams illustrate the pin alignment of each connector.

Legend :

A/D	-	Analog input
A.GND	-	Analog ground
D/A	-	Analog output
D/O	-	Digital output
D/I	-	Digital input
D.GND	-	Digital and power supply ground
CLK	-	Clock input for the 8253
GATE	-	Gate input for the 8253
OUT	-	Signal output of the 8253
VREF	-	Voltage reference

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Connector 1 (CN1) - Analog Input (Single-ended channels)

A/D	0	1	2	A.GND
A/D	1	3	4	A.GND
A/D	2	5	6	A.GND
A/D	3	7	8	A.GND
A/D	4	9	10	A.GND
A/D	5	11	12	A.GND
A/D	6	13	14	A.GND
A/D	7	15	16	A.GND
A/D	8	17	18	A.GND
A/D	9	19	20	A.GND

Connector 2 (CN2) - Analog Output

A/D	10	1	2	A.GND
A/D	11	3	4	A.GND
A/D	12	5	6	A.GND
A/D	13	7	8	A.GND
A/D	14	9	10	A.GND
A/D	15	11	12	A.GND
D/A	1	13	14	A.GND
D/A	2	15	16	A.GND
V.REF	1	17	18	A.GND
V.REF	2	19	20	A.GND

Connector 3 (CN3) - Digital Output

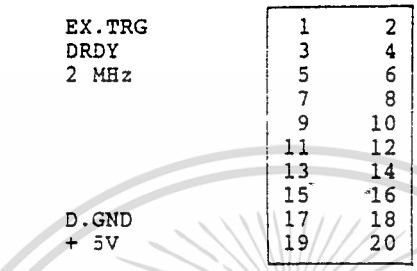
D/O	0	1	2	D/O 1
D/O	2	3	4	D/O 3
D/O	4	5	6	D/O 5
D/O	6	7	8	D/O 7
D/O	8	9	10	D/O 9
D/O	10	11	12	D/O 11
D/O	12	13	14	D/O 13
D/O	14	15	16	D/O 15
D.GND		17	18	D.GND
+ 5V		19	20	+ 12V

Connector 4 (CN4) - Digital Input

D/I	0	1	2	D/I 1
D/I	2	3	4	D/I 3
D/I	4	5	6	D/I 5
D/I	6	7	8	D/I 7
D/I	8	9	10	D/I 9
D/I	10	11	12	D/I 11
D/I	12	13	14	D/I 13
D/I	14	15	16	D/I 15
D.GND		17	18	D.GND
+ 5V		19	20	+ 12V

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Connector 5 (CN5) - Counter



2-4 Hardware Installation

```
************************  WARNING  ***************************
**   TURN OFF your PC power supply whenever installing or   **
**   removing the PCL-812 or connecting and disconnecting   **
**   cables.                                                **
**************************************************************
```

Installing the card in your computer :

1. Turn the computer off. Turn the power off to any peripheral devices (such as printers and monitors).

2. Disconnect the power cord and any other cables from the back of the computer. Turn the system unit so the back of the unit faces you.

3. Remove the system unit cover (refer to your computer users guide if necessary).

4. Locate the expansion slots at the rear of the unit and choose any unused slot.

5. Remove the screw that secures the expansion slot cover to the system unit (save the screw to secure the interface card retaining bracket).

6. Carefully grasp the upper edge of the PCL-812 card. Align the hole in the retaining bracket with the hole on top of the expansion slot, and align the gold striped edge connector with the expansion slot socket. Press the board firmly into the socket.

7. Replace the screw in the expansion slot retaining bracket.

8. Attach necessary accessories (e.g. 20 pin flat cable or connector adapter, etc.) to the interface card based on your

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

application requirements.

9. Replace the system unit cover. Connect the cables you removed in step 2. Turn the computer power on.

The hardware installation is now complete, proceed to install the software driver.

2-5 Software Installation

A floppy diskette containing utility software is included with each PCL-812 to minimize your application programming work and support the PCL-812 calibration. The utility programs include:

1. A comprehensive I/O driver routine for A/D, D/A, Digital I/O and Counter applications. It is written in assembly language and accessed through the CALL statement of BASICA and Quick BASIC.

2. Demonstration programs.

3. Calibration program.

It is strongly recommended that you make a working copy from the master copy diskette and save the master copy in safe place. You may use the DOS COPY or DISKCOPY commands to copy the diskette files to another floppy disk or use the COPY command to copy the files to a hard disk. Refer to your MS-DOS users manual for details of how to make backup copies of your diskette.

Since the operation of I/O driver routines, demonstration programs and calibration program are covered in Chapter 5,6,7,8,9 and Appendix D, we will only explain how to load the PCL-812 driver routines to the PC memory segment in this section.

The PCL-812 I/O driver is written in assembly language and compiled into a machine language program called 812BAS.BIN. The actual size of the program is more than 4K bytes. There are two possible ways of loading the I/O driver into memory, your selection depends on the memory size available on you PC.

1. The first method is loading the machine language driver immediately after the normal BASIC work space area. This approach is more complicated than the other approach, because it requires close verification of the free bytes available in the BASIC work space area, and calculation of the driver routine starting address.

   Steps involved (in sequence) are :

   a. Check the free bytes available in the BASIC work space

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

area.

b. If the BASIC free work space does not exceed 60,000 bytes, it is recommended that you use the BASICA WS (work space) command to reduce the BASIC work space area and reserve sufficient space for the driver routines.

c. Identify the segment address containing BASIC.

d. Calculate the end address of the BASIC work space from the position the driver program is loaded.

e. Load the binary program 812BAS.BIN using the BASIC "BLOAD" command.

A sample program is provided as following.

```
110  'LOAD 812BAS.BIN DRIVER TO BASIC AREA
120  CLEAR 57344!  'SET BASIC HIGHEST WORK SPACE WITHIN 56K
130  DEF SEG=0
140  SG=256*PEEK(&H511)+PEEK(&H510)   'GET BASIC SEGMENT
150  SG=SG+57344!/16
160  DEF SEG=SG
170  BLOAD"812BAS.BIN",0
180  'END OF DRIVER LOADING
```

2. The second method is loading the machine language driver to a memory segment which is independent from the BASIC work area. This approach is easier to use, but it requires more available memory space on your PC.

The steps involved (in sequence) are :

a. Locate a memory segment with at least 5K bytes free at its beginning.

b. Load the binary program 812BAS.BIN with the BASIC "BLOAD" command from the beginning of that memory segment.

A sample program is provided as following.

```
110  'LOAD 812BAS.BIN DRIVER TO AN OUTSIDE AREA
120  DEF SEG=&H5000   'DEFINE OUTSIDE AREA
130  BLOAD"812BAS.BIN", 0
140  'END OF DRIVER LOADING
```

Since the driver is located in an area outside the BASIC work area, the driver should be linked by a DEF command immediately before each driver FUNCTION being called. An example is provided.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
150  'INITIALIZE DRIVER USING FUNCTION 0
160  DAT%(0)=&H220
170  DAT%(1)=4
180  DAT%(2)=1
190  PCL812=0
200  ERR%=0
210  FUNC%=0
220  DEF SEG=&H5000
230  CALL PCL812(FUNC%,DAT%(0), ERR%)
240  'END OF DRIVER INITIALIZATION
```

Loading the driver is not a difficult task, but remember that the PCL-812 driver requires at least 5K bytes memory size which should not be used by any other programs simultaneously.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

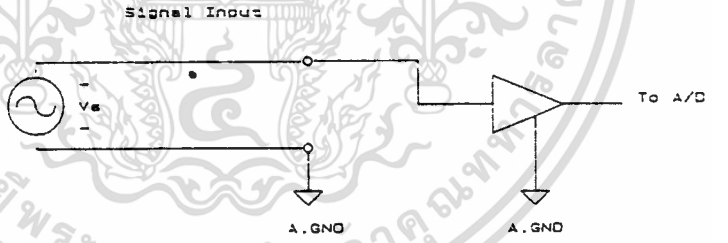
CHAPTER 3 SIGNAL CONNECTION

INTRODUCTION

Correct signal connection is one of the most important steps to assure your application system will send and receive data correctly. Since most data acquisition applications involve voltage, correct signal connection will avoid costly damage to your personal computer and hardware devices. This chapter provides some useful information on signal connection in different types of data acquisition applications.

3-1 Analog Input Connection

The PCL-812 supports a 16 single-ended analog input configuration. The single-ended configuration has only one signal wire for each channel. The voltage to be measured is the voltage of this wire referred to the common ground. A signal source without a local ground is also called a "floating source". It is fairly simple to connect a single ended channel to a floating signal source. A standard wiring diagram is illustrated below :



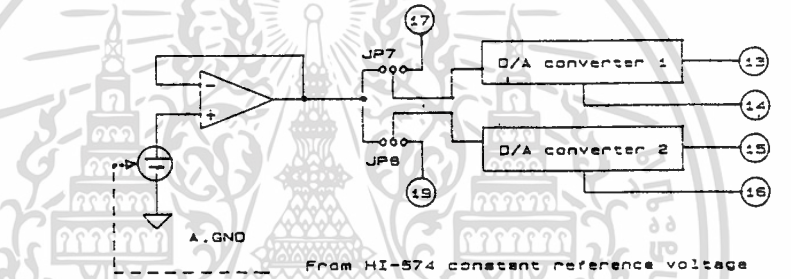
3-2 Expanding Analog Inputs

You may expand any or all of the PCL-812 A/D input channels through sub-multiplexers. The PCLD-789 Amplifier and Multiplexer daughter board in the PC-LabCard series is specifically designed for multiplexing applications. Each PCLD-789 can multiplex 16 differential inputs to one A/D input channel. Up to eight PCLD-789 can be cascaded to one PCL-812 providing a total of 128 channels. Complete operation information of using the PCL-812 with the PCLD-789 is covered in the PCLD-789 users manual. For PCLD-789 product information, please contact your local PC-LabCard sales representative.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3-3 Analog Output Connection

PCL-812 provides two D/A output channels. You may use the PCL-812's internally generated precision -5V reference to generate 0 to +5V D/A output range. You may create other D/A output ranges through external references. The maximum reference input range is +/- 10V and maximum output scaling is +/- 10V. The PCL-812 Connector # 2 is used for D/A signals. Some important areas in D/A signal connections are input reference, D/A outputs and analog ground. The PCL-812 D/A's should be connected as illustrated below:



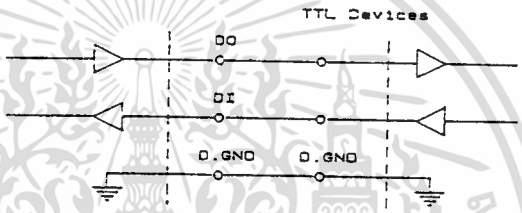
Legend (Connector 2 pin number):

13 = D/A 1 D/A outputs
15 = D/A 2 D/A outputs
14 = Analog ground
16 = Analog ground
17 = V.REF1
19 = V.REF2

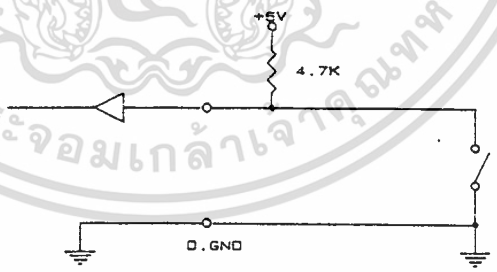
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3-4 Digital Signal Connection

The PCL-812 has 16 digital input and 16 digital output channels. The digital I/O levels are TTL compatible. To transmit or receive digital signal to/from other TTL devices, the connection is:



To receive an OPEN/SHORT signal from a switch or relay, a pull up resistor must be added to ensure the high level when open.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CHAPTER 4 REGISTER STRUCTURE AND FORMAT

INTRODUCTION

The PCL-812 requires 16 consecutive addresses in I/O space. The most important issue in programming the PCL-812 is understanding the meaning of the 16 registers addressable from the selected I/O port base address. A summary map of the functions of each address and the data format of each register are given in the following sections.

4-1 I/O Port Address Map

The following table shows the location of each register and driver relative to the base address, and its usage.

Location	Read	Write
Base + 0	Counter 0	Counter 0
+ 1	Counter 1	Counter 1
+ 2	Counter 2	Counter 2
+ 3	N/U	Counter control
+ 4	A/D low byte	CH1 D/A low byte
+ 5	A/D high byte	CH1 D/A high byte
+ 6	D/I low byte	CH2 D/A low byte
+ 7	D/I high byte	CH2 D/A high byte
+ 8	N/U	Clear interrupt request
+ 9	N/U	N/U
+10	N/U	MUX scan channel
+11	N/U	PCL-812 Control
+12	N/U	Software A/D trigger
+13	N/U	D/O low byte
+14	N/U	D/O high byte
+16	N/U	

* N/U = Not Used

4-2 A/D Data Registers

The A/D data registers use address BASE +4 and +5.

Data Format :

1. A/D Low byte and Channel number.

	D7	D6	D5	D4	D3	D2	D1	D0
BASE +4	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

2. A/D High byte.

BASE +5	D7	D6	D5	D4	D3	D2	D1	D0
	0	0	0	DRDY	AD11	AD10	AD9	AD8

LEGEND :

AD11 to AD0 - Analog to digital data. AD0 is the least significant byte (LSB) and AD11 is the most significant byte (MSB) of the A/D data.

DRDY - Data ready signal.

4-3 Mux Scan Register

The MUX scan register is a write only register using address BASE +10. The low nybble provides the scan channel number. The MUX multiplexer switches to the new channel when writing to this register.

Data Format :

BASE +10	D7	D6	D5	D4	D3	D2	D1	D0
Scan channel	X	X	X	X	CL3	CL2	CL1	CL0

Legend :

CL3 to CL0 - Scan channel number.

4-4 Digital I/O Registers

The PCL-812 offers 16 digital input channels and 16 digital output channels. These I/O channels use the input address BASE +6 and BASE +7. The output ports are at BASE +13 and BASE +14. The data format of each port is as follows:

Data Format :

BASE +6 (read port)	D7	D6	D5	D4	D3	D2	D1	D0
D/I low byte	DI7	DI6	DI5	DI4	DI3	DI2	DI1	DI0
BASE +13 (write port)	D7	D6	D5	D4	D3	D2	D1	D0
D/O low byte	DO7	DO6	DO5	DO4	DO3	DO2	DO1	DO0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

BASE +7 (read port) D/I high byte	D7 DI15	D6 DI14	D5 DI13	D4 DI12	D3 DI11	D2 DI10	D1 DI9	D0 DI8
BASE +14 (write port) D/O high Byte	D7 DO15	D6 DO14	D5 DO13	D4 DO12	D3 DO11	D2 DO10	D1 DO9	D0 DO8

4-5 D/A Output Registers

The D/A output registers are write registers using address BASE +4, +5, +6 and +7.

Data Format :

BASE +4	D7	D6	D5	D4	D3	D2	D1	D0
D/A #1 low byte	DA7	DA6	DA5	DA4	DA3	DA2	DA1	DA0
BASE +5	D7	D6	D5	D4	D3	D2	D1	D0
D/A #1 high byte	X	X	X	X	DA11	DA10	DA9	DA8
BASE +6	D7	D6	D5	D4	D3	D2	D1	D0
D/A #2 low byte	DA7	DA6	DA5	DA4	DA3	DA2	DA1	DA0
BASE +7	D7	D6	D5	D4	D3	D2	D1	D0
D/A #2 high byte	X	X	X	X	DA11	DA10	DA9	DA8

LEGEND :

DA11 to DA0 - Digital to analog data. DA0 is the least significant byte (LSB) and DA11 is the most significant byte (MSB) of the D/A data.

4-6 PCL-812 Control Register

The PCL-812 control register is a write-only register using address BASE +11. This register provides information on the operating modes of the PCL-812.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

BASE +7 (read port) D/I high byte	D7 DI15	D6 DI14	D5 DI13	D4 DI12	D3 DI11	D2 DI10	D1 DI9	D0 DI8
BASE +14 (write port) D/O high Byte	D7 DO15	D6 DO14	D5 DO13	D4 DO12	D3 DO11	D2 DO10	D1 DO9	D0 DO8

4-5 D/A Output Registers

The D/A output registers are write registers using address BASE +4, +5, +6 and +7.

Data Format :

BASE +4	D7	D6	D5	D4	D3	D2	D1	D0
D/A #1 low byte	DA7	DA6	DA5	DA4	DA3	DA2	DA1	DA0
BASE +5	D7	D6	D5	D4	D3	D2	D1	D0
D/A #1 high byte	X	X	X	X	DA11	DA10	DA9	DA8
BASE +6	D7	D6	D5	D4	D3	D2	D1	D0
D/A #2 low byte	DA7	DA6	DA5	DA4	DA3	DA2	DA1	DA0
BASE +7	D7	D6	D5	D4	D3	D2	D1	D0
D/A #2 high byte	X	X	X	X	DA11	DA10	DA9	DA8

LEGEND :

DA11 to DA0 - Digital to analog data. DA0 is the least significant byte (LSB) and DA11 is the most significant byte (MSB) of the D/A data.

4-6 PCL-812 Control Register

The PCL-812 control register is a write-only register using address BASE +11. This register provides information on the operating modes of the PCL-812.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

Data Format :

BASE +11	D7	D6	D5	D4	D3	D2	D1	D0
PCL-812 CNTL status	X	X	X	X	X	S2	S1	S0

A. Under internal trigger condition

S2	S1	S0		
0	0	0	:	Disable Software & Pacer trigger.
0	0	1	:	Enable Software trigger only
0	1	0	:	Enable Pacer trigger using DMA transfer only
1	1	0	:	Enable Pacer trigger using program transfer or interrupt transfer. If using pacer trigger and program transfer, the jumper JP4 must be set to the "X" position, as below:

```
IRQ  2 3 4 5 6 7 X
JP4  o o o o o o o
     o o o o o o o
```

B. Under external trigger source condition

S2	S1	S0		
0	0	X	:	Enable External trigger only.
0	1	0	:	Enable External trigger using DMA transfer Data only
1	1	0	:	Enable program data transfer and interrupt transfer using external trigger source.

* Note : Set up trigger jumper JP1 on EXT before use each trigger mode, as below :

```
     JP1
INT   o
TRG  [o]
EXT  [o]
```

4-7 Programmable Interval Timer-Counter Registers

The four registers located at address BASE 0, 1, 2, and 3 are used for the Intel 8253 programmable timer/counter. Please refer to Chapter 8 or the 8253 product literature for detailed application information.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CHAPTER 5 A/D CONVERSION

INTRODUCTION

This chapter provides a complete explanation of how to use the PCL-812 A/D conversion functions. It covers A/D data format, input range selection, status register, MUX scan setting, trigger modes and data transfer in the first five sections. The last section gives step by step implementation guidelines with sample programs on different A/D operations.

5-1 A/D Data Format and Status register

When the PCL-812 performs 12 bit A/D conversions, an 8 bit register is not big enough to accommodate all 12 bits of data. Therefore A/D data are stored in two registers located at address BASE +4 and +5. The A/D low byte data are in the positions D0 (AD0) through D7 (AD7) of BASE +4 and high byte data are in the positions D0 (AD8) through 3(AD11) of BASE +5. The least significant data is AD0 and the most significant data is AD11. The A/D channel number from which the conversion data was derived is available at BASE +10 position D0 (C0) to D3 (C3).

The data formats of the A/D data registers are :

A/D Low byte and Channel number.

BASE +4	D7	D6	D5	D4	D3	D2	D1	D0
	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0

A/D High byte.

BASE +5	D7	D6	D5	D4	D3	D2	D1	D0
	0	0	0	DRDY	AD11	AD10	AD9	AD8

5-2 MUX Setting

Data Format :

BASE +10	D7	D6	D5	D4	D3	D2	D1	D0
	X	X	X	X	CL3	CL2	CL1	CL0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการ
ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5-3 Trigger Mode

The PCL-812 A/D conversions can be triggered in any one of three ways - software trigger, on-board programmable pacer or external pulse trigger.

1. The software trigger is controlled by the application program issued software command. This trigger mode normally is not used in high speed A/D applications due to the limitations of the applications program execution time.

2. The PCL-812 uses the INTEL 8253 programmable interval timer/counter. Counters 1 and 2 of the INTEL 8253 are configured to be a pacer to offer A/D converter trigger pulses with precise periods in the pacer trigger mode. The pacer output of the PCL-812 is between 0.5 MHz and 35 minutes per pulse. Chapter 8. covers the details of using the INTEL 8253 timer/counter. The pacer trigger mode is ideal for interrupt and DMA data transfer which normally used in A/D applications requiring a higher conversion speed.

3. The PCL-812 direct external trigger pulses are controlled through EXT.TRG (Connector 5 pin #1). This type of trigger mode is mostly used in A/D applications requiring A/D conversions not periodically but conditionally e.g. thermocouple temperature control.

5-4 A/D Data Transfer

There are three possible ways to perform the PCL-812 A/D data transfer - by program control, interrupt routine or DMA.

1. The program control data transfer uses the polling concept. After the A/D converter has been triggered, the application program checks the End of Conversion (EOC) bit of the A/D status register. If the EOC is detected, the converted data is moved from the A/D data register to computer memory by application program control.

2. In interrupt routine transfer, data is transferred from the A/D data registers to a previously defined memory segment by the interrupt routine handler. At the end of each conversion, the EOC signal generates an interrupt which enables the interrupt handler routine to perform the transfer. The interrupt control bit and interrupt level selection in the PCL-812 control register (BASE +11) must be specified before the use of interrupt routine. A write action to the A/D status register address (BASE +8) resets the PCL-812 interrupt request and re-enables the PCL-812 interrupt.

3. Direct memory access (DMA) transfer moves the A/D data from the PCL-812 hardware device to the PC system memory without operation of the system CPU. DMA is very useful in high speed data transfer, but it is complicated to operate. The DMA level selection jumper, JP5 and JP6, and the DMA enable bit in the PCL-812 control register as well as the 8237 DMA controller registers must be set up before performing DMA operations. It is recommended that users use the PCL-812 driver FUNCTION6 to perform DMA operation. For more information regarding the 8237 DMA controller and the PCL-812 DMA operations, please read Chapter 10.

5-5 How to Execute an A/D Conversion

You may execute A/D operations with a program writing all I/O port instructions directly, or by a program utilizing the PCL-812 driver routines. It is suggested that you use the driver routines in your program. This will make your programming job easier and enhance the program performance. In the last section of this chapter, step by step implementation procedures for different A/D operations are introduced.

1. To perform pacer trigger and program control data transfer without the PCL-812 driver.

   Step 1 : Set the input channel by specifying the Mux scan range.

   Step 2 : Write the pacer trigger mode to the control register.

   Step 3 : Wait for the EOC by reading the A/D high byte register (BASE +5) EOC bit.

   Step 4 : Read data from A/D converter by reading the A/D data registers (BASE +5 and +4). Must first read high byte.

   Step 5 : Data conversion by converting the binary A/D data to an integer.

   The demonstration program (DEMO01.BAS) on the software diskette performs this operation.

2. To perform software trigger and program control data transfer with PCL-812 driver routines.

   Step 1 : Load the 812BAS driver program to a memory segment.

   Step 2 : Use FUNCTION 0 to initialize driver.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Step 3 : Use FUNCTION 1 to set input channel range.

Step 4 : Use FUNCTION 3 to perform single A/D conversion.

The demonstration program (DEMO02.BAS) on the software diskette performs this operation.

3. To perform pacer trigger and direct to array data transfer with PCL-812 driver.

Step 1 : Load the 812BAS driver program to a memory segment.

Step 2 : Use FUNCTION 0 to initialize driver.

Step 3 : Use FUNCTION 1 to set input channel range.

Step 4 : Use FUNCTION 17 to set pacer trigger rate.

Step 5 : Use FUNCTION 4 to perform N A/D conversions and transfer data to array.

The demonstration program (DEMO03.BAS) in the software diskette performs this operation.

4. To perform pacer trigger and interrupt data transfer with PCL-812 driver.

Step 1 : Load the 812BAS driver program to a memory segment.

Step 2 : Use FUNCTION 0 to initialize driver.

Step 3 : Use FUNCTION 1 to set input channel range.

Step 4 : Use FUNCTION 17 to set pacer trigger rate.

Step 5 : Use FUNCTION 5 to perform N A/D conversions and save to memory with interrupt transfer in background operation.

Step 6 : Use FUNCTION 8 to check interrupt operation status in foreground operation.

Step 7 : Use FUNCTION 9 to transfer data from memory to array.

Step 8 : Display A/D data from array.

Step 9 : Use FUNCTION 7 to disable the interrupt activity.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

The demonstration program (DEMO04.BAS) in the software diskette performs this operation.

5. To perform pacer trigger and DMA data transfer with PCL-812 driver.

Step 1 : Load the 812BAS driver program to a memory segment.

Step 2 : Use FUNCTION 0 to initialize driver.

Step 3 : Use FUNCTION 1 to set input channel range.

Step 4 : Use FUNCTION 17 to set pacer trigger rate.

Step 5 : Use FUNCTION 6 to perform N A/D conversions and save to memory with DMA transfer in background operation.

Step 6 : Use FUNCTION 8 to check DMA operation status in foreground operation.

Step 7 : Use FUNCTION 9 to transfer data from memory to array.

Step 8 : Display A/D data from array.

Step 9 : Use FUNCTION 7 to disable the DMA activity.

The demonstration program (DEMO05.BAS) on the software diskette performs this operation.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CHAPTER 6 D/A CONVERSION

6-1 General Information

The PCL-812 provides 2 D/A channels which use 2 double buffered 12 bit multiplying D/A converters. The D/A registers are write registers using address BASE +4, +5, +6 and +7. DA0 is the least significant byte (LSB) and DA11 is the most significant byte (MSB) of the D/A data. A summary of each register's data format is given below:

BASE +4	D7	D6	D5	D4	D3	D2	D1	D0
D/A #1 low byte	DA7	DA6	DA5	DA4	DA3	DA2	DA1	DA0
BASE +5	D7	D6	D5	D4	D3	D2	D1	D0
D/A #1 high byte	X	X	X	X	DA11	DA10	DA9	DA8
BASE +6	D7	D6	D5	D4	D3	D2	D1	D0
D/A #2 low byte	DA7	DA6	DA5	DA4	DA3	DA2	DA1	DA0
BASE +7	D7	D6	D5	D4	D3	D2	D1	D0
D/A #2 high byte	X	X	X	X	DA11	DA10	DA9	DA8

When writing to the D/A channels, please note that the least significant byte should be written first. It is then temporarily held by a register in the D/A and not released to the output. After the most significant byte is written, the low byte and high byte are added and passed to the D/A converter. This double buffering process protects the D/A data integrity through a single step update.

The PCL-812 provides an internal precision fixed -5V reference. If this voltage is used as D/A input reference, the D/A output range will be 0 to +5V. You may select other external DC or AC sources as the D/A reference inputs. The maximum reference voltage is -/+ 10V and the maximum D/A output range is +/- 10V.

Connector 2 supports all D/A signal connections. The pin assignment of Connector 2 is described in Chapter 2, Section 2-3. Section 3-3 covers D/A signal connection, and gives an illustration of a wiring diagram.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6-2 D/A Applications

A variety of D/A operations can be supported by your PCL-812 For example the PCL-812 can function as a digital attenuator by inputing variable AC or DC references, or can be used to generate arbitrary waveform outputs. As with the PCL-812 A/D programming, the D/A functions can be performed in two ways - by using the PCL-812 driver functions in the application program, or writing I/O instructions to the registers directly. There are two PCL-812 driver functions performing D/A related operations.

FUNCTION 15 - Performs output on single D/A channel.

FUNCTION 16 - Performs output on both D/A channels.

A commonly used D/A application is clearly illustrated in the a demonstration program on your diskette. The demonstration program, DEMO08.BAS, on the PCL-812 software diskette provides a complete explanation of how to use FUNCTION 15 and FUNCTION 16. It sends D/A outputs to either a single D/A channel or both D/A channels.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CHAPTER 7 DIGITAL INPUT AND OUTPUT

The PCL-812 provides 16 digital input channels and 16 digital output channels. These I/O channels use the input at address BASE +6 and BASE +7 and output registers at BASE +13 and BASE +14. A summary of each register's data format is listed below.

BASE +6 (read port)	D7	D6	D5	D4	D3	D2	D1	D0
D/I low byte	DI7	DI6	DI5	DI4	DI3	DI2	DI1	DI0
BASE +13 (write port)	D7	D6	D5	D4	D3	D2	D1	D0
D/O low byte	DO7	DO6	DO5	DO4	DO3	DO2	DO1	DO0
BASE +7 (read port)	D7	D6	D5	D4	D3	D2	D1	D0
D/I high byte	DI15	DI14	DI13	DI12	DI11	DI10	DI9	DI8
BASE +14 (write port)	D7	D6	D5	D4	D3	D2	D1	D0
D/O high byte	DO15	DO14	DO13	DO12	DO11	DO10	DO9	DO8

It is fairly straight forward to use your PCL-812 digital input and output functions. Attention should be paid to the pin assignments of Connector 3 (digital output) and Connector 4 (digital input). Section 3-4 offers some suggestions about digital signal connections.

There are two PCL-812 driver functions performing digital input or output operations :

FUNCTION 21 - Write digital output to channel DO0 through DO15.

FUNCTION 22 - Read digital input from channel DI0 through DI15.

The demonstration program, DEMO11.BAS, on the PCL-812 software diskette provides a good example of using the driver FUNCTION 21 and 22 to perform digital input and output functions.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CHAPTER 8 PROGRAMMABLE INTERVAL TIMER/COUNTER

8-1 The Intel 8253

The PCL-812 uses the INTEL 8253 programmable interval timer/counter version 5. The 8253 is a very popular timer/counter device consisting of three independent 16 bit down counters. Each counter has a clock input, control gate and an output. It can be programmed to have a count from 2 up to 65535.

The maximum clock input frequency is 5 MHz for version 5 of the 8253. The PCL-812 provides a 2 MHz input frequency through an on-board crystal oscillator.

Counters 1 and 2 are cascaded and operated in fixed divider configuration. The counter 2 input is connected to the 2 MHZ frequency and the output of counter 2 is connected to the input of counter 1. The output of Counter 1 is internally configured to provide a trigger pulse to the A/D converter. Counter 0 is not reserved by the PCL-812 for any internal use, and you may access counter 0 through Connector 5. Please refer to Section 2-3 for details of the connector 5 pin assignment.

8-2 Counter Read/Write and Control Registers

The 8253 programmable interval timer uses four registers at address BASE +0, 1, 2 and 3. The function of each register is:

BASE +0	Counter 0 Read/Write
BASE +1	Counter 1 Read/Write
BASE +2	Counter 2 Read/Write
BASE +3	Counter Control Word

Since the 8253 counter uses a 16 bit structure, each read/write data is split into the least significant byte and the most significant byte. It is important to ensure your read/write operations are in pairs and keep track of the byte order.

The data format of the control register is :

	D7	D6	D5	D4	D3	D2	D1	D0
BASE +3	SC1	SC0	RW1	RW0	M2	M1	M0	BUD

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Legend :

SC1 & SC0 - Select Counter.

SC1	SC0	Counter
0	0	0
0	1	1
1	0	2
1	1	Illegal

RW1 & RW0 - Select the Read/Write operation.

RW1	RW0	Operation
0	0	Counter latch
0	1	Read/Write LSB
1	0	Read/Write MSB
1	1	Read/Write LSB first, then MSB.

M2, M1 and M0 - Select the Operation Mode.

M2	M1	M0	Mode
0	0	0	0 - Interrupt on terminal count
0	0	1	1 - Programmable one shot
X	1	0	2 - Rate generator
X	1	1	3 - Square wave rate generator
1	0	0	4 - Software triggered strobe
1	0	1	5 - Hardware triggered strobe

BUD - Select Binary or BUD Counting.

BUD	Type
0	Binary Counter 16-bits
1	Binary Coded Decimal (BUD) Counter (4 Decades)

If the modules is set to be binary, the count can be any number from 0 up to 65535. If the modules is set to be BUD (binary coded decimal), the count can be set as any number from 0 to 9999.

8-3 Counter Operating Modes

8-3-1 Mode 0 - Interrupt on Terminal Count

The output will initially be low after setting this mode operation. After the count is loaded into the selected count register, the output will remain low and the counter will start to

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

count. When the terminal count is reached, the output will go high and remain high until the selected counter is reloaded with the mode, or a new count is loaded. The counter continues to decrement after the terminal count has been reached. Rewriting a counter register during counting generates the following results:

1. Write the first byte stops the current counting.

2. Write the second byte starts the new count.

8-3-2 Mode 1 - Programmable One-Shot

The output will go low on the count following the rising edge of the gate input. The output will go high on the terminal count. If a new count value is loaded while the output is low, it will not affect the duration of the one-shot pulse until the succeeding trigger. The current count can be read at any time without affecting the one-shot pulse. The one-shot is retriggerable, thus the output will remain low for the full count after any rising edge of the gate input.

8-3-3 Mode 2 - Rate Generator (Divide by N Counter)

The output will be low for one period of the input clock. The period from one output pulse to the next is equal to the number of input counts in the counter register. If the counter register is reloaded between output pulses, the present period will not be affected, but the subsequent period will reflect the value.

The gate input, when low, will force the output high. When the gate input goes high, the counter will start from the initial count. Therefore the gate input can be used to synchronize the counter.

When this mode is set, the output will remain high until the count register is loaded and the output can also be synchronized by software.

8-3-4 Mode 3 - Square Wave Rate Generator

Similar to Mode 2, except that the output will remain high until one half the count has been completed (for even numbers) and go low for the other half of the count. This is accomplished by decrementing the counter by two on the falling edge of each clock pulse. When the counter reaches the terminal count, the state of the output is changed and the counter is reloaded with the full count and the whole process is repeated.

If the count is odd and the output is high, the first clock pulse

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(after the count is loaded) decrements the count by one. Subsequent clock pulses decrement the count by two. After timeout, the output goes low and the full count is reloaded. The first clock pulse (following the reload) decrements the counter by three. Subsequent clock pulses decrement the count by two until timeout. Then the whole process is repeated. In this way, if the count is odd, the output will be high for (N+1)/2 counts and low for (N-1)/2 counts.

8-3-5 Mode 4 - Software Triggered Strobe

After the mode is set, the output will be high. When the count is loaded, the counter will begin counting. On terminal count, the output will go low for one input clock period, then will go high again.

If the count register is reloaded during counting, the new count will be loaded on the next CLK pulse. The count will be inhibited while the GATE input is low.

8-3-6 Mode 5 - Hardware Triggered Strobe

The counter will start counting after the rising edge of the trigger input and will go low for one clock period when the terminal count is reached. The counter is retriggerable.

8-4 Counter Operations

8-4-1 Read/Write Operations

For each counter, the type of read/write operation, operating mode and counter type all must be properly specified in the control byte and the control byte must be written before the initial count is written.

Since the control byte register and all three counter read/write registers have separate addresses and each control byte specifies the counter it applies to (by SC1 & SC0), no instructions on the operating sequence are required. Any programming sequence following the 8253 conventions is acceptable.

There are three types of counter operation - read/load LCB, read/load MSB and read/load LSB followed by MSB. It is important to ensure your read/write operations are in pairs and keep track of the byte order.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

8-4-2 Counter Latch Operation

It is often desirable to read the value of a counter without disturbing the count in progress. Usually the method used is the counter latch command method which allows the user to read the latched count value of the selected counter.

The 8253 supports counter latch operations in two ways. The first way is set the RW1 & RW0 to be (0,0) which latches the count of the selected counter in a 16 bit hold register. This method has the advantage of operating several counters at the same time. A subsequent rad operation on the selected counter will retrieve the held value.

The second approach is to perform the latch operation by setting the MODE Register.

MODE Register for Latching Count.

A0,A1 = 11

D7	D6	D5	D4	D3	D2	D1	D0
SC1	SC0	0	0	X	X	X	X

SC1,SC0 - specify counter to be latched.

D5,D4 - 00 designates counter latching operation.

X - don't care

The same limitation applies to this mode of reading the counter as the previous method. That is, it is mandatory to complete the entire read operation as programmed. This command has no effect on the counter's mode.

8-5 Counter Applications

The 8253 programmable interval timer/counter on your PCL-812 interface card is a very useful device. In this section, two of the most popular usages of the 8253 timer/counter are introduced.

1. If you are using the A/D function of the PCL-812, the 8253 can be programmed to serve as a pacer to generate A/D trigger pulse. The demonstration program, DEMO12.BAS.on the PCL-812 software diskette provides an example of using the driver FUNCTION 17 to generate an A/D trigger pulse.

2. Counter 0 of the 8253 is not committed to any internal use. Users may configure Counter 0 to perform any 8253 supported functions e.g.to act as a square wave generator. The

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

demonstration program, DEMO13.BAS, on the PCL-812 software diskette provides an example of using Counter 0 to count external events or generate a square wave.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CHAPTER 9 PROGRAMMING

9-1 Introduction to The PCL-812 Programming

There are two ways to accomplish the PCL-812 functions in an application program. The first method is using the PCL-812 driver routines in the application program and let the driver handle the I/O port interfaces. Another method is to write the I/O port instructions in the application program directly. Both programming methods are discussed in detail in the following sections.

9-1-1 Programming without The PCL-812 Driver Routines

This programming method requires a good understanding of basic register structure, data format, and the PCL-812 hardware. You also need more knowledge of the PC interrupt handler routine and DMA controller. It requires more effort to handle the I/O address directly, but you have flexibility in dealing with very special application requirements. If you use this programming method, you can use assembly language and some other high level languages e.g. BASIC or C, which support I/O input and output instructions.

The demonstration program, DEMO01.BAS, on the PCL-812 software diskette provides an example of writing the I/O port instructions directly. This program performs software trigger A/D and program control data transfer.

9-1-2 Programming with The PCL-812 Driver Routines

Your PCL-812 comes with assembly language written driver routines. The functions covered by these driver routines include applications governing A/D, D/A, Digital I/O and Programmable Interval Timer. They also covers different type of A/D applications i.e. A/D with interrupt transfer, DMA transfer.

The PCL-812 driver routines are designed to work with BASIC language programming. You can access the driver functions through the BASIC CALL statement. Below, we will explain how to use the CALL statement in the PCL-812 application programming.

CALL statement format :

```
CALL PCL812 (FUNC%, DAT%(0), ERR%)
```

Legend :

PCL812 : A variable specifying the memory offset of the starting address of the PCL812 driver routine from the most recently defined current segment. It is recommended to load the PCL812 driver routines from the beginning of a memory segment and set PCL812 = 0. We use PCL812 as the variable name for easy recognition.

FUNC% : A variable indicating the driver function to be executed.

DAT%(0) : A variable specifying the starting location of the entire data array.

ERR% : A variable containing the return message code.

Application note :

1. The naming of the variables is not critical and you may change the names. However, the parameter position is very important, you must follow the specified sequence.

2. All parameters must be integer variables. You may use array variables for the DAT%(X) variables, but make sure you have dimensioned the array variables before calling them.

3. No parameters can be represented by a constant directly; it must be previously defined before the CALL statement. No arithmetic functions are permitted within the CALL statement.

4. Some functions, such as FUNCTION 4 and 9, require the dimension of a receiving array to store the A/D converted data. Please note that after assigning the pointer to the receiving array, no other new simple variables are introduced before the CALL statement. The following two examples indicate the correct and incorrect handling of a receiving array.

Correct :

```
500 DIM ARY%(599)                'CREATE 600'S ARRAY LOCATIONS
510 DAT%(0)=600                  'NUMBER OF CONVERSIONS
520 DAT%(1)=VARPTR(ARY%(0))      'GET ARRAY START ADDRESS
530 DAT%(2)=1                    'TRIGGER SOURCE FROM PACER
540 FUNC%=4
550 CALL PCL812(FUNC%, DAT%(0), ERR%)
```

Incorrect :

```
500 DIM ARY%(599)                'CREATE 600'S ARRAY LOCATIONS
510 DAT%(0)=600                  'NUMBER OF CONVERSIONS
520 DAT%(1)=VARPTR(ARY%(0))      'GET ARRAY START ADDRESS
```


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

```
530 DAT%(2)=1                    'TRIGGER SOURCE FROM PACER
540 FUNC%=4
545 NEW-VARIABLE = XXX           'INTRODUCE NEW VARIABLE
550 CALL PCL812(FUNC%, DAT%(0), ERR%)
```

9-2 PCL-812 Driver Routines

There are twenty-two PCL-812 driver functions. Each function covers an important area of the use of the PCL-812. The following is a list of all functions.

Function	Description
FUNCTION 0	Initialize the PCL-812 driver routines by specifying the base address, interrupt and DMA levels.
FUNCTION 1	Set the multiplexer scan range.
FUNCTION 2	Reserved.
FUNCTION 3	Perform a software triggered single A/D conversion.
FUNCTION 4	Perform programmable interval timer or external trigger pulse initiated N A/D conversions.
FUNCTION 5	Perform programmable interval timer or external trigger pulse initiated N A/D conversions and save the collected data to memory with interrupt transfer.
FUNCTION 6	Perform programmable interval timer or external trigger pulse initiated N A/D conversions and save the collected data to memory with DMA transfer.
FUNCTION 7	Terminates FUNCTION 5, 6 initiated interrupt or DMA operations.
FUNCTION 8	Provide status information on FUNCTION 5, 6 initiated interrupt or DMA operations.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FUNCTION 9	Convert FUNCTION 5 or 6 retrieved data from memory segment to array variable.
FUNCTION 10	Set up the Counter 0 operation mode.
FUNCTION 11	Write Counter 0 data.
FUNCTION 12	Read Counter 0 data.
FUNCTION 13	Reserved.
FUNCTION 14	Reserved.
FUNCTION 15	Write data to one of the two D/A channels.
FUNCTION 16	Write data to both D/A channels.
FUNCTION 17	Set A/D pacer trigger rate by specifying Counter 1 and Counter 2 division ratio.
FUNCTION 18	Reserved.
FUNCTION 19	Reserved.
FUNCTION 20	Reserved.
FUNCTION 21	Write data to digital output channels DO0 through DO15.
FUNCTION 22	Read the state of digital input channels DI0 through DI15.

9-2-1 FUNCTION 0 - Initialization

Purpose :

This function initializes the PCL-812 base address, interrupt level and DMA level according to the parameters specified by the user. It also initializes the following functions :

1. Disables all external triggers, interrupt and DMA functions.
2. Resets the PCL-812 control and timer counter enable registers.

This function must be performed before any other functions and only one initialization is necessary in each operation session.

Parameter :

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FUNC% - Function. The value is 0.

DAT%(0) - Entry : I/O port address. The default value is Hex 220.

DAT%(1) - Entry : Interrupt level. Valid values are 2 to 7

DAT%(2) - Entry : DMA level. Valid values are 1 or 3.

ERR% - Return message.

Message List :

ERR% = 0 Good return.
= 2 Invalid function number.
= 3 Invalid I/O port.
= 4 Invalid interrupt level.
= 5 Invalid DMA level.

Example :

```
100  PORT%=&H220                'GIVEN I/O PORT START ADDRESS
110  DAT%(0)=PORT%              'GET I/O PORT START ADDRESS
120  DAT%(1)= 3                 'USE IRQ 3
130  DAT%(2)= 1                 'USE DRQ 1
140  PCL812 = 0                 'DRIVER OFFSET
150  ERR% = 0                   'INITIAL ERROR VALUE
160  FUNC% = 0
170  CALL PCL812(FUNC%, DAT%(0), ERR%)
180  IF ERR%<>0 THEN PRINT "INSTALLATION FAILED !":STOP
```

9-2-2 FUNCTION 1 - Set MUX Scan Range

Purpose :

This function sets the multiplexer scan range. If the current scan range doesn't meet the next A/D conversion range requirements, this function must be performed prior to calling any A/D conversion functions i.e. FUNCTION 3,4,5,6.

Parameter :

FUNC% - Function. The value is 1.

DAT%(0) - Entry : Start scan channel. Valid values are 0 to 15.

DAT%(1) - Entry : Stop scan channel. Valid values are 0 to 15.

ERR% - Return Message.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Message List :

ERR% = 0 Good return.
= 1 No driver initialization.
= 2 Invalid function number.
= 7 Out of channel range.

Remark : In the single-ended configuration mode, the start scan channel number can be greater than the end of scan channel number. If your application requires continuous conversions on a single channel, just set the start scan and stop scan channel number to be identical.

Example :

```
200  DAT%(0)=3                    'START SCAN CHANNEL
210  DAT%(1)=5                    'STOP SCAN CHANNEL
220  FUNC%=1
230  CALL PCL812(FUNC%,DAT%(0),ERR%)
240  IF ERR%<>0 THEN PRINT "SET SCAN CHANNEL FAILED !":STOP
```

9-2-3 FUNCTION 3 - Perform Single A/D Conversion

Purpose :

This function performs a software triggered single A/D conversion.

Parameter :

FUNC% - Function. The value is 3.

DAT%(0) - Entry : Not significant.
Return : Converted A/D data (- 2048 to + 2047 for bipolar).

DAT%(1) - Entry : Not significant.
Return : Channel number of data derived from.

DAT%(2) - Entry : Not significant.
Return : Next channel number.

ERR% - Return Message.

Message List :

ERR% = 0 Good return.
= 1 No driver initialization.
= 2 Invalid function number.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

= 9 A/D converter hardware error (no EOC).

Example :

```
400  FUNC% = 3
410  CALL PCL812(FUNC%, DAT%(0),ERR%)
420  IF ERR%<>0 THEN PRINT "SOFTWARE TRIGGER FAILED !":STOP
430  PRINT USING"SCAN CHANNEL=##
     READING=####";DAT%(1),DAT%(0)
440  A$=INKEY$:IF A$="" GOTO 400   'RE-TRIG
```

9-2-4 FUNCTION 4 - Perform N A/D Conversions and Save to Array Variable

Purpose :

This function performs programmable interval timer or external trigger pulse initiated N A/D conversions. The A/D conversion and data saving are performed as a foreground operation which means the CPU is fully occupied until the completion of all A/D conversions.

If you want to perform the A/D conversion as a background operation, please use FUNCTION 5 or 6 which allow you to collect and process data simultaneously. It is recommended to use this function for medium speed (less than 2.5 KHz) applications.

Since FUNCTION 4 performs the A/D operations in the foreground, if the computer is hung up due to erroneous operation or incorrect parameter settings, like insufficient number of trigger pulses to cover the required number of conversions, you may hit any key on the keyboard to terminate the operation.

If an external trigger is used, triggered pulses are connected to EX.TRG. Positive edges start A/D conversions.

Parameter :

FUNC% - Function. The value is 4.

DAT%(0) - Entry : Number of conversions required. Valid values are 1 through N (array dimension must > or = N-1).

DAT%(1) - Entry : Array pointer at which the converted data can be anywhere in an array, but make sure the available array size can cover the number of conversions specified. The format is DAT%(1) = VARPTR(ARY%(x)).

DAT%(2) - Entry : Trigger source. Use 0 for external

trigger and 1 for pacer trigger.

DAT%(3) - Entry : Channel array to store the corresponding channel number of each conversion. The format is DAT%(3)=VARPTR(CHARY%(0)).

ERR% - Return Message.

Message List :

ERR% = 0 Good return.
= 1 No driver initialization.
= 2 Invalid function number.
= 11 Invalid number of conversions (0 or negative).
= 19 Invalid trigger source code (not 0 or 1).

Remark : You should define a receiving array of the proper size to store the converted data before calling this function. Do not specify any new variables after assigning the pointer to the receiving array.

Example :

```
500  DIM ARY%(599),CHARY%(599) 'CREATE 600'S ARRAY LOCATIONS
510  DAT%(0)=600               'NUMBER OF CONVERSIONS
520  DAT%(1)=VARPTR(ARY%(0))   'GET DATA ARRAY START ADDRESS
530  DAT%(2)=1                 'TRIGGER SOURCE FROM PACER
535  DAT%(3)=VARPTR(CHARY%(0)) 'GET CH. ARRAY START ADDRESS
540  FUNC%=4
550  CALL PCL812(FUNC%, DAT%(0), ERR%)
560  IF ERR%<>0 THEN PRINT "DIRECT TRIGGER FAILED!":STOP
570  FOR I=0 TO 599
580  PRINT ARY%(I), CHARY%(I)  'PRINT A/D DATA
590  NEXT I
```

9-2-5 FUNCTION 5 - Perform N A/D Conversions and Save to Memory with Interrupt Transfer

Purpose :

This function performs programmable interval timer or external trigger pulse initiated N A/D conversions and saves the collected data to memory with interrupt transfers.

At the end of each conversion, an interrupt handler routine is invoked to transfer the data to a defined memory segment. The next conversion is continued as soon as the interrupt handler is established.

The interrupts are repeated continuously under the re-cycle mode, or terminated automatically under the non-recycle mode. Please note that the interrupt handler cannot be re-established before

it is terminated. Therefore if you are using the re-cycle mode, you must run FUNCTION 7 to terminate the interrupt before calling the FUNCTION 5 again.

If an external trigger is used, triggered pulses are connected to EX.TRG, positive edges start A/D conversions.

This function is good for medium speed (less than 3 KHz) applications. In many aspects it is very similar to FUNCTION 6 which uses the DMA transfer to save data at very high speed.

Parameter :

FUNC% - Function. The value is 5.

DAT%(0) - Entry : Number of conversions required. Valid values are 1 through 32,767.

DAT%(1) - Entry : Memory segment for data store. The interrupt transferred data are loaded from the beginning of the segment. Be aware that your memory segment can save up to 32,767 bytes of conversion data. Also be careful about problems of overlapping of the memory segment.

DAT%(2) - Entry : Trigger source.
0 = External trigger input.
1 = Programmable interval timer.

DAT%(3) - Entry : Non-recycle/Recycle mode.
0 = Non-recycle. The interrupts are terminated after completing the number of conversions.
1 = Recycle. The conversions are repeated continuously until the calling of FUNCTION 7.

ERR% - Return Message.

Message List :

ERR%	=	0	Good return.
	=	1	No driver initialization.
	=	2	Invalid function number.
	=	8	Invalid non-recycle or recycle mode code.
	=	11	Invalid number of conversions (0 or negative).
	=	19	Invalid trigger source code (not 0 or 1).
	=	20	Interrupt or DMA is active.

Remark : The following functions perform FUNCTION 5 related activities.

FUNCTION 7 - Terminate interrupts.
FUNCTION 8 - Check interrupt status.
FUNCTION 9 - Convert data from memory to array variable.

Example :

```
500  DAT%(0)=500          'NUMBER OF CONVERSIONS
510  DAT%(1)=&H4000       'SEGMENT OF MEMORY TO STORE DATA
520  DAT%(2)=1            'TRIG SOURCE FROM PACER
530  DAT%(3)=0            'ONE CYCLE SAMPLE DATA
540  FUNC%=5
550  CALL PCL812(FUNC%, DAT%(0), ERR%)
560  IF ERR%<>0 THEN PRINT "INTERRUPT MODE FAILED!":STOP
570  'NEXT USING FUNCTION 9 TO TRANSFER DATA TO ARRAY
```

9-2-6 FUNCTION 6 - Perform N A/D Conversions and Save to Memory with DMA Transfer

Purpose :

This function performs programmable interval timer or external trigger pulse initiated N A/D conversions and saves the collected data to memory with DMA transfer. The DMA transfer uses the PC 8237 DMA Controller.

The DMA transfers are repeated continuously under the re-cycle mode, or terminated automatically under the non-recycle mode when N conversions are completed. Please note that once DMA or interrupt handler is enabled, you may not reestablish the DMA or interrupt handler before it is terminated. Therefore, you must run FUNCTION 7 to disable previous FUNCTION 5 or FUNCTION 6 recycle mode operation before you can call FUNCTION 6 again.

If an external trigger is used, triggered pulses are connected to EX.TRG, positive edges start A/D conversions.

This function provides you with a convenient way to perform very high speed A/D. The standard PCL-812 can accomplish 30 KHz sampling rate.

Parameter :

FUNC% - Function. The value is 6.

DAT%(0) - Entry : Number of conversions required. Valid values are 1 through 32,767.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DAT%(1) - Entry : Memory segment for data store. The DMA transferred data are loaded from the beginning of the segment. Be aware that your memory segment can save up to 32,767 bytes of conversion data. Also be careful about the problem of memory segment overlapping.

DAT%(2) - Entry : Trigger source.
0 = External trigger input.
1 = Programmable interval timer.

DAT%(3) - Entry : Non-recycle/Recycle mode.
0 = Non-recycle. The DMA is terminated after completing the specified number of conversions.
1 = Recycle. This mode is connected to the DMA automatic initialization, therefore the A/D conversions are repeated continuously until the calling of FUNCTION 7.

ERR% - Return Message.

Message List :

ERR% =	0	Good return.
=	1	No driver initialization.
=	2	Invalid function number.
=	8	Invalid non-recycle or recycle mode code.
=	11	Invalid number of conversions (0 or negative).
=	19	Invalid trigger source code (not 0 or 1).
=	20	Interrupt or DMA is active.
=	21	DMA page wrap around.

Remark : The following functions perform FUNCTION 6 related activities.

FUNCTION 7 - Terminate interrupts.
FUNCTION 8 - Check interrupt status.
FUNCTION 9 - Convert data from memory to array variable.

Due to the limitations of DMA page registers, A/D data saved to the segment can't exceed the page boundary. An error message (code 21) is generated by the driver for this page wrap around problem. A good practice is to start the segment from the page boundary.

Example :

```
500  DAT%(0)=500              'NUMBER OF CONVERSIONS
```

```
510  DAT%(1)=&H4000          'SEGMENT OF MEMORY TO STORE DATA
520  DAT%(2)=1               'TRIG SOURCE FROM PACER
530  DAT%(3)=0               'ONE CYCLE SAMPLE DATA
540  DAT%=6
550  CALL PCL812(FUNC%, DAT%(0), ERR%)
560  IF ERR%<>0 THEN PRINT "DMA MODE FAILED!":STOP
570  'NEXT USING FUNCTION 9 TO TRANSFER DATA TO ARRAY
```

9-2-7 FUNCTION 7 - Disable Interrupt and DMA Activity

Purpose :

This function terminates FUNCTION 5, 6, 18, 20 initiated interrupt or DMA operations.

Parameter :

FUNC% - Function. The value is 7.

DAT%(0) - Entry: Not significant.

ERR% - Return Message.

Message List :

ERR% = 0 Good return.
= 1 No driver initialization.
= 2 Invalid function number.

Example :

```
800  FUNC%=7
810  CALL PCL812(FUNC%,DAT%(0),ERR%)
820  IF ERR%<>0 THEN PRINT "DISABLE FAILED!":STOP
```

9-2-8 FUNCTION 8 - Check Interrupt and DMA Status

Purpose :

This function provides status information on FUNCTION 5, 6 initiated interrupt or DMA operations.

Parameter :

FUNC% - Function. The value is 8.

DAT%(0) - Entry : Not significant.
Return : Operation mode.
0 = No Interrupt/DMA operation
1 = DMA

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

2 = FUNCTION 5 Interrupt

DAT%(1) - Entry : Not significant.
Return : Number of current conversion.

ERR% - Return Message.

Message List :

ERR% = 0 Good return.
= 1 No driver initialization..
= 2 Invalid function number.

Example :

```
600   FUNC%=8
610   CALL PCL812(FUNC%,DAT%(0),ERR%)
620   IF ER%<>0 THEN PRINT "READ A/D STATUS FAILED!":STOP
630   LOCATE 5,2
640   PRINT "FUNCTION     MODE-";DAT%(0)
650   PRINT "CONVERSION STATUS-";DAT%(1)
660   PRINT "CURRENT    NUMBER-";DAT%(2)
670   AS=INKEY$:IF AS="" GOTO 600
```

9-2-9 FUNCTION 9 - Convert Data from Memory to Array Variable

Purpose :

This function converts FUNCTION 5 and 6 retrieved data from the memory segment to array variables. It is an ideal tool to support on-line real-time applications if you set the FUNCTION 5 or 6 recycle mode as background operation and use this function to support foreground display or analysis operations.

The great flexibility on specifying parameters allows you to move any number of conversions from any source memory segment location to any location in an array variable.

Parameter :

FUNC% - Function. The value is 9.

DAT%(0) - Entry : Number of conversions to be moved. Valid value is 1 to 32767.

DAT%(1) - Entry : Source memory segment.

DAT%(2) - Entry : Beginning conversion number. Valid number is 0 to 32767.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DAT%(3) - Entry : Data array to store moved data. The starting position can be anywhere in an array, but make sure the available array size can cover the number of conversions to be moved. The format is DAT%(3) = VARPTR(DARR%(x)).

DAT%(4) - Entry : Channel array to store the corresponding channel number of each conversion. The format is DAT%(4) = VARPTR(CARR%(x))

ERR% - Return Message.

Message List :

ERR% =	0	Good return.
=	1	No driver initialization.
=	2	Invalid function number.
=	17	Invalid beginning conversion number.
=	18	Invalid number of conversions to be moved in DAT%(0).

Remark :

1. Please allocate sufficient array variables to accommodate the transferred data.

2. Do not specify any new variables after assigning the pointer to the receiving array (i.e. VARPTR(DARR%(x)) and VARPTR(CARR%(x)) since the declaring of new variable relocates the assigned array.

Example :

```
700   DIM DTA%(499),CHL%(499) 'CREATE ARRAY LOCATION
710   DAT%(0)=500             'NUMBER OF DATA TRANSFER
720   DAT%(1)=&H4000          'SEGMENT OF MEMORY
730   DAT%(2)=0               'TRANSFER DATA FROM FIRST DATA
740   DAT%(3)=VARPTR(DARR%(0)) 'GET ARRAY START ADDRESS
750   DAT%(4)=VARPTR(CARR%(0)) 'GET ARRAY START ADDRESS
760   FUNC%=9
770   CALL PCL812(FUNC%,DAT%(0),ERR%)
780   IF ERR% <> 0 THEN PRINT"TRANSFER DATA FAILED!":STOP
790   FOR I=0 TO 499
800   PRINT USING"SCAN CHANNEL=## READING=####";
      CHL%(I),DTA%(I)
810   NEXT  I
```

9-2-10 FUNCTION 10 - Set-up Counter 0 Operation Mode

Purpose :

This function establishes the Counter 0 operation mode. There are six operation modes available to Counter 0. For detailed descriptions on the programmable interval timer/counter, please refer to Chapter 8.

Parameter :

FUNC% - Function. The value is 10.

DAT%(0) - Entry : Operation mode. Valid value is 0 to 5.
0 = Interrupt on terminal count.
1 = Programmable one shot.
2 = Rate generator.
3 = Square wave generator.
4 = Software triggered strobe.
5 = Hardware triggered strobe.

ERR% - Return Message.

Message List :

ERR% =	0	Good return.
=	1	No driver initialization.
=	2	Invalid function number.
=	12	Invalid counter operation mode.

Example :

```
300   DAT%(0)=3          'SET 8253 MODE-SQUARE WAVE GEN.
310   FUNC%=10
320   CALL PCL812(FUNC%,DAT%(0),ERR%)
330   IF ERR% <> 0 THEN PRINT "SET CTR0 MODE FAILED!":STOP
```

9-2-11 FUNCTION 11 - Write Counter 0 Data

Purpose :

This function loads the Counter 0 data. The variable can be loaded ranging from 0 to 65535 because Counter 0 is a 16 bit device. However, the actual storage is done in integer variables (+32767 to -32768) in order to save memory space and minimize operation time.

Parameter :

FUNC% - Function. The value is 11.

```
     DAT%(0) -  Entry :  Counter data.  Valid value is integer
                         variables from +32767 to -32768 for
                         actual variable of 0 to 65535.

     ERR%    -   Return Message.

Message List :

     ERR% = 0  Good return.
          = 1  No driver initialization.
          = 2  Invalid function number.

Example :

     400  DAT%(0)=300              'LOAD COUNT DATA
     410  FUNC%=11
     420  CALL PCL812(FUNC%,DAT%(0),ERR%)
     430  IF  ERR%<>0  THEN  PRINT  "LOAD  CTR0  COUNT  DATAIP6,13
          FAILED!":STOP


9-2-12  FUNCTION 12 - Read Counter 0 Data

Purpose :

This function reads the Counter 0 data.  The returned data is in
integer variable format (+32767 to -32768).  This corresponds to
actual values from 0 to 65535.

Parameter :

     FUNC%  -  Function.  The value is 12.

     DAT%(0) -  Entry :  Read operation mode.
                         0 =  Straight  read  operation  for  static
                              counter data.
                         1 =  Latch read operation for dynamic counter
                              data (counter is counting).

     ERR%    -  Return Message.

Message List :

     ERR% = 0  Good return.
          = 1  No driver initialization.
          = 2  Invalid function number.
          = 16 Invalid read operation mode(DAT% not = 0 or 1).

Example :
```

```
500  DAT%(0)=1                    'SET LATCH READ OPERATION
510  FUNC%=12
520  CALL PCL812(FUNC%,DAT%(0), ERR%)
530  IF ERR%<>0 THEN PRINT "READ CTR0 FAILED!":STOP
540  PRINT"READING=":DAT%(1)
550  A$=INKEY$:IF A$=""GOTO 500
```

9-2-13 FUNCTION 15 – Set Output on Single D/A Channel

Purpose :

This function writes data to one of the two D/A channels.

Parameter :

FUNC% – Function. The value is 15.

DAT%(0) – Entry : D/A channel number. Valid value is 1, or 2.

DAT%(1) – Entry : D/A data. Valid value is 0 to 4095.

ERR% – Return Message.

Message List :

```
ERR% = 0  Good return.
     = 1  No valid initialization.
     = 2  Invalid function number.
     = 14 Invalid D/A data (not 0 to 4095).
     = 15 Invalid D/A channel number (not 1 or 2).
```

Example :

```
300  DAT%(0)=0                    'SELECT D/A CHANNEL 0
310  DAT%(1)=2048                 'D/A VOLTS=Verf*2048/4096
320  FUNC%=15
330  CALL PCL812(FUNC%,DAT%(0),ERR%)
340  IF ERR%<>0  THEN PRINT "SET SINGLE D/A FAILED!":STOP
```

9-2-14 FUNCTION 16 – Set Output on Both D/A Channels

Purpose :

This function writes data to both D/A channels.

Parameter :

FUNC% – Function. The value is 16.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DAT%(0) - Entry : D/A channel 1 data. Valid value is 0 to 4095.

DAT%(1) - Entry : D/A channel 2 data. Valid value is 0 to 4095.

ERR% - Return Message.

Message List :

ERR% = 0 Good return.
= 1 No driver initialization.
= 2 Invalid function number.
= 14 Invalid D/A data (not 0 to 4095).

Example :

```
300   DAT%(0)=1024          'SET D/A #1 VOLTS=Vref*1024/4096
310   DAT%(1)=2346          'SET D/A #2 VOLTS=VREF*2346/4096
320   FUNC%=16
330   CALL PCL812(FUNC%,DAT%(0),ERR%)
340   IF ERR%<>0 THEN PRINT "SET D/A #1 AND D/A #2
      FAILED!":STOP
```

9-2-15 FUNCTION 17 - Set Pacer Trigger Rate

Purpose :

This function sets the A/D pacer trigger rate by specifying the divisor of Counter 1 and Counter 2. The pacer trigger rate is calculated as:

Pacer rate = Input clock rate / (C2 X C1)

where C2 = Counter 2 divisor
C1 = Counter 1 divisor

and the input clock rate is fixed at 2MHz.

Parameter :

FUNC% - Punction. The value is 17.

DAT%(0) - Entry : Counter 2 divisor. Valid value is 2 to 65535.

DAT%(1) - Entry : Counter 1 divisor. Valid value is 2 to 65535.

ERR% - Return Message.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Message List :

```
ERR% =  0     Good return.
     =  1     No driver initialization.
     =  2     Invalid function number.
     =  10    Invalid counter division ratio (0 or 1).
```

Example :

```
400   DAT%(0)=50                    'CTR2 DIVISOR DATA
410   DAT%(1)=20                    'CTR1 DIVISOR DATA
420   'PACER TIMER RATE=2MHz/(50*20)
430   FUNC%=17
440   CALL ABSOLUTE(FUNC%,DAT%(0),ERR%)
450   IF ERR%<>0 THEN PRINT "SET PACER FAILED!":STOP
```

9-2-16 FUNCTION 21 - Write Digital Output DO0 - DO15

Purpose :

This function writes data to digital output channels DO0 through DO15.

Parameter :

FUNC% - Function. The value is 21.

DAT%(0) - Entry : Output data for DO0 - DO7. Valid values are 0 to 255.

DAT%(1) - Entry : Output data for DO8 - DO15. Valid values are 0 to 255.

ERR% - Return Message.

Message List :

```
ERR%  =  0     Good return.
      =  1     No driver initialization.
      =  2     Invalid function number.
      =  13    Invalid output dat.  (DAT% not 0 to 255).
```

Example :

```
300   DAT%(0)=255    'LOW BYTE DIGITAL OUTPUT DATA (0-255)
310   DAT%(1)=200    'HIGH BYTE DIGITAL OUTPUT DATA (0-255)
320   FUNC%=21
330   CALL PCL812(FUNC%,DAT%(0),ERR%)
340   IF ERR%<>0 THEN PRINT "WRITE DIGITAL DATA
      FAILED!":STOP
```


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

9-2-17 FUNCTION 22 - Read Digital Input DO0 - DO15

Purpose :

This function reads the state of digital input channels DI0 through DI15.

Parameter :

FUNC% - Function. The value is 22.

DAT%(0) - Entry : Not significant.
Return : Input data for DI0 to DI7. Valid value is 0 to 255.

DAT%(1) - Entry : Not significant.
Return : Input data for DI8 to DI15. Valid value is 0 to 255.

ERR% - Return Message.

Message List :

ERR%	=	0	Good return.
	=	1	No driver initialization.
	=	2	Invalid function number.

Example :

```
400     FUNC%=22
410     CALL PCL812(FUNC%,DAT%(0),ERR%)
420     IF ERR%<>0 THEN PRINT "READ DIGITAL DATA
        FAILED!":STOP
430     DIG=DAT%(1)*256+DAT%(0)
440     PRINT "DECIMAL DATA=###### HEXDECIMAL=\
        \";DIG,HEX$(DIG)
```

9-3 Driver Return Code Listing

The following is a complete list of return messages generated by the driver functions.

Code	Description
0	Good return, no error. Used in all FUNCTIONS.
1	No driver initialization. Invalid use of other FUNCTION before initializing the driver. Used in all FUNCTIONS except FUNCTION 0.

2 Invalid FUNCTION. Valid function numbers are 0 to 22. Used in all FUNCTIONS.

3 Invalid I/O port address range. Used in FUNCTION 0.

4 Invalid interrupt level. Valid levels are 2 to 7. Used in FUNCTION 0.

5 Invalid DMA level. Valid levels are 1 or 3. Used in FUNCTION 0.

6 RESERVED

7 Out of single-ended input channel range. Valid values are 0 to 15. Used in FUNCTION 1.

8 Invalid non-recycle or recycle mode code. Valid values are 0 and 1. Used in FUNCTION 5 and 6.

9 A/D converter hardware error, no end of conversion (EOC). Used in FUNCTION 3.

10 Invalid counter division ratio, Valid values are 2 to 65535. Used in FUNCTION 17.

11 Invalid number of conversions. Valid values are 1 to 32767. Used in FUNCTION 4, 5, and 6.

12 Invalid counter operation mode. Valid values are 0 to 5. Used in FUNCTION 10.

13 Invalid digital output data. Used in FUNCTION 21.

14 Invalid D/A data. Valid values are 0 to 4095. Used in FUNCTION 15 and 16.

15 Invalid D/A channel number. Valid values are 1 or 2. Used in FUNCTION 15.

16 Invalid counter read operation mode. Valid values are 0 or 1. Used in FUNCTION 12.

17 Invalid beginning conversion number. Valid values are 0 to 32767. Used in FUNCTION 9.

18 Invalid number of conversions transferred. Valid values are 1 to 32767. Used in FUNCTION 9.

19 Invalid trigger source code. Valid values are 0 or 1. Used in FUNCTION 4, 5, and 6.

20 Interrupt or DMA operation is active. Used in FUNCTION

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5, and 6.

21 DMA page wrap around. Used in FUNCTION 6.

9-4 Programming Examples

To assist your application programming, we have prepared 9 demonstration programs on the software diskette enclosed in the PCL-812 package. Each program covers an important application area and comes with a complete program description. The use of each program is discussed in the related Chapters. All demonstration programs are written in BASIC. Apart from DEMO01 which writes the I/O instructions to the port directly, all other programs use the PCL-812 driver routines. The following is a list of all programs and their applications.

Program Name	Function
DEMO01.BAS	Performs software trigger A/D and program control data transfer without PCL-812
DEMO02.BAS	Performs software trigger A/D and program control data transfer.
DEMO03.BAS	Performs pacer trigger A/D and direct to array data transfer.
DEMO04.BAS	Performs pacer trigger A/D and interrupt data transfer.
DEMO05.BAS	Performs pacer trigger A/D and DMA dat transfer.
DEMO08.BAS	Performs outputs to either single D/A channel or both D/A channels.
DEMO11.BAS	Performs digital input/output on channel DI0 to DI15 and DO0 to DO15.
DEMO12.BAS	Programs the Counter 1 and 2 to generate A/D trigger pulse.
DEMO13.BAS	Program the Counter 0 to count external event or generate square wave.

CHAPTER 10 DIRECT MEMORY ACCESS (DMA) OPERATION

INTRODUCTION

Direct memory access (DMA) improves system performance by allowing external devices to directly transfer information to or from the system memory without operation of the system CPU. The PCL-812 is designed with A/D data DMA transfer capability. This feature significantly improves system performance during high speed A/D applications.

10-1 Introduction to The PC 8237 DMA Controller

DMA is controlled by the 8237 DMA controller chip on the PC's system board. It has four priority direct memory access channels. Channel 0 is reserved by the PC system to perform dynamic RAM refresh. Channel 2 is always assigned to support floppy disk operations. Channel 3 is normally used by the hard disk operations. Channel 1 is not reserved for any internal operations and is available for user applications.

Each channel has two control signals associated with it. The DMA request signal (DRQ) triggers a DMA operation, and the DMA acknowledge signal (DACK) authorizes the DMA to start data transfer.

In addition to four DMA channels, the 8273 DMA chip has four operating modes (single, demand, block and cascade) and four control registers. These registers are

1. Operation mode register (set operation mode).
2. Address register (specify memory segment starting address).
3. Word count register (specify the number of transfers).
4. Initialization register (enable and disable DMA channels).

Please note that all four registers must be properly set up before the DMA operation can abe requested.

10-2 How to Use DMA Transfer with The PCL-812

DMA transfer is a powerful but complicated operation. Different subjects regarding the DMA transfer have been covered in many chapters of this manual. The following is a summarization of how to use DMA transfer with the PCL-812. For detailed explanation on different subjects, please refer to the specific chapters.

1. During hardware configuration, check your PC DMA channel availability (level 1 or level 3) and set the PCL-812 jumper

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

JP5, JP6 (jumper) accordingly.

2. If you choose to use the PCL-812 driver routine FUNCTION 6 to support the DMA transfer programming, simply provide the required data e.g. number of conversions, memory segment address etc. to the driver. And the driver will take care of the complicated operations for you.

3. If you choose to conduct your own DMA operation, you need to have a solid understanding of the PC, 8237 DMA controller and the PCL-812 device. To complete a DMA transfer, make sure you have covered the following operations.

   a. Initialize 8237 DMA controller register and page register.

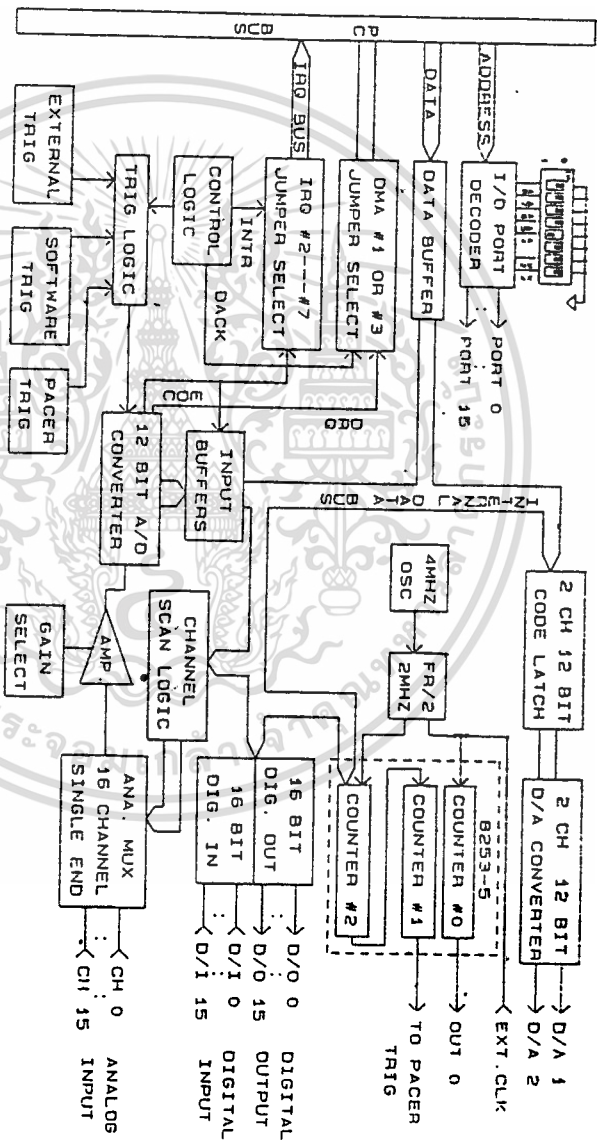
   b. Set JP5 and JP6 to correct DMA channel.

   c. Send DMA enable and trigger source data to the PCL-812 control register located at address BASE +11.

   d. Set up external trigger pulse or pacer trigger rate.

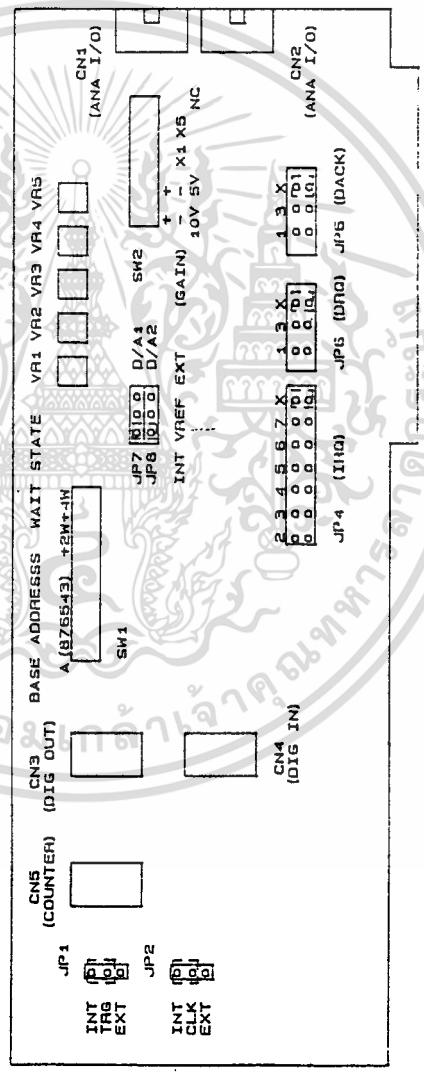
   e. Enable trigger source to start A/D conversion.

APPENDIX A PCL-812 BLOCK DIAGRAM



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

APPENDIX B PCL-812 CONNECTOR, SWITCH AND VR LOCATIONS



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

APPENDIX C PC I/O PORT ADDRESS MAP

I/O Address Range (Hex)	Function
000-1FF	Base system
200	Reserved
201	Game control
202-277	Reserved
278-27F	Second printer port
280-2F7	Reserved
2F8-2FF	COM2
300-377	Reserved
378-37F	First printer port
380-3AF	Reserved
3B0-3BF	Mono Disp/Print adapter
3C0-3CF	Reserved
3D0-3DF	Color/Graphics
3E0-3EF	Reserved
3F0-3F7	Floppy disk drive
3F8-3FF	COM1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

APPENDIX D CALIBRATION

In data acquisition and control, it is important to constantly calibrate your measurement device to maintain its accuracy. A calibration program, CALB.BAS, is provided on the PCL-812 software diskette to assist your calibration work.

The calibration program is written in BASIC and utilizes the PCL-812 driver routines. To use it, simply load the program into BASIC and run the program. The default I/O port address setting in the program is Hex 220. If address Hex 220 had been occupied by another device, you may either replace that device with the PCL-812 on address Hex 220 or modify the address setting in the calibration program. It is strongly recommended that you don't change the program unless you thoroughly understand the PCL-812 driver functions and the calibration program.

Since the calibration program requires the PCL-812 driver program, 812BAS.BIN, to operate, when you are loading the calibration program to BASIC, please ensure that the 812BAS.BIN program is also available on the same floppy diskette or hard disk.

Once the calibration program has been loaded and executed, it uses the graphic display and prompts to guide you through the calibration process.

In addition to the calibration program, it is necessary to have a 5 1/2 digit multimeter and a voltage calibrator or very stable and noise free DC voltage source to perform the calibration. A slot extension card will make your access to the VRs easier. The PC-LabCARD product PCL-755A (Slot Extension Card) is an ideal product to support the slot extension.

D.1 PCL-812 VR Assignment

There are 5 variable resistors (VR) on the PCL-812 to allow you making accurate adjustment on A/D and D/A channels. The location of each VR is indicated in Appendix B (PCL-812 Connector, Switch and VR Locations). The function of each VR is listed below:

VR1 : D/A 1 gain adjustment
VR2 : D/A 2 gain adjustment
VR3 : A/D gain adjustment (+/-5V range)
VR4 : 5 Times gain adjustment (+/-1V range)
VR5 : A/D offset adjustment

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

D.2 A/D Calibration

Since the PCL-812 provides many A/D input ranges, the calibration on one A/D range may not be accurate on another range. It is suggested that you calibrate the A/D whenever you change the input range.

The calibration program will prompt you to specify the following items :

1. Channel configuration (single-ended)
2. Input range setting
3. Channel number

It then leads you to adjust the A/D offset and A/D gain. It will also ask you to perform the common mode rejection ration adjustment, if you are using the differential channel configuration.

D.3 D/A Calibration

The D/A input reference should be connected to the channel to be calibrated. You may use the on-board - 5V reference or other external references. The full scale gain of each D/A channel are adjusted through different VR. The reading should be 4.9997V for the full scale gain.