ระบบควบคุมแบบกระจายส่วน ขนาดเล็ก

SMALL SCALE DCS

โดย

DIGITAL INPUT

นาย นเรศ ตันพิชัย

นาย พนมกร ลิมปาวิภากร

ANALOG OUTPUT

นาย ธีระ จันทร์อนุวงศ์

นาย อำนวยสิทธิ์ ศรียาพันธ์

DIGITAL OUTPUT

นาย ศิริชัย คล่องการพานิช

นาย ปัญญา อัศวพนิต

ANALOG INPUT

นาย วรเวช อิทธิธานันท์

PID UNIT

นาย จักรกฤษณ์ ดัจฉกรณ์

นาย ดุษฎี ลีลาสมาน

นาย วีรชาติ ตริไตรตรอง

DATA LINKER

นาย ปัญญา อาจหาญ

นาย คนัย ทักษนนท์

นาย เจษฎา สีหะเนิน

ANALOG I/O CONTROL UNIT & SYSTEM SOFTWARE

นาย สิชล ชุมชนะ

นาย วัชระ โบราณวรรณ์

นาย นพดล สุวรรณเจษฎา

วัน เดือน ปี 29 ก.ย. 2541

เลขทะเบียน 038016

เลขเรียกหนังสือ T 39036 พ 2539

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตร์บัณฑิต

สาขาวิศวกรรมการวัดคุมทางอุตสาหกรรม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2539

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ระบบควบคุมแบบกระจายส่วน ขนาดเล็ก

SMALL SCALE DCS

โดย

DIGITAL INPUT

นาย นเรศ ตันพิชัย

นาย พนมกร ลิมปาวิภากร

ANALOG OUTPUT

นาย ธีระ จันทร์อนุวงศ์

นาย อำนวยสิทธิ์ ศรียาพันธ์

DIGITAL OUTPUT

นาย ศิริชัย คล่องการพานิช

นาย ปัญญา อัศวพนิต

ANALOG INPUT

นาย วรเวช อิทธิชานันท์

PID UNIT

นาย จักรกฤษณ์ ตัจฉกรณ์

นาย ดุษฎี ลีลาสมาน

นาย วีรชาติ ตริไตรตรอง

DATA LINKER

นาย ปัญญา อาจหาญ

นาย ดนัย ทักษนนท์

นาย เจษฎา สีหะเนิน

ANALOG I/O CONTROL UNIT & SYSTEM SOFTWARE

นาย สิชล ชุมชนะ

นาย วัชระ โบราณวรรณ์

นาย นพดล สุวรรณเจษฎา

อาจารย์ที่ปรึกษา

ผศ.ธนิตย์ ตรีสุวรรณวัฒน์

อ.ประสิทธิ์ จุลเสรีวงศ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาควิชาเทคโนโลยีการวัดคุมทางอุตสาหกรรม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบควบคุมแบบกระจายส่วน ขนาดเล็ก

SMALL SCALE DCS

ผู้จัดทำ

DIGITAL INPUT

นาย นเรศ ตันพิชัย

นาย พนมกร ลิมปาวิภากร

ANALOG OUTPUT

นาย ธีระ จันทร์อนุวงศ์

นาย อำนวยสิทธิ์ ศรียาพันธ์

DIGITAL OUTPUT

นาย ศิริชัย คล่องการพานิช

นาย ปัญญา อัศวพนิต

ANALOG INPUT

นาย วรเวช อิทธิชานันท์

PID UNIT

นาย จักรกฤษณ์ ตัจฉกรณ์

นาย ดุษฎี ลีลาสมาน

นาย วีรชาติ ตริไตรตรอง

DATA LINKER

นาย ปัญญา อาจหาญ

นาย ดนัย ทักษนนท์

นาย เฉษฎา สีทะเนิน

ANALOG I/O CONTROL UNIT & SYSTEM SOFTWARE

นาย สิชล ชุมชนะ

นาย วัชระ โบราณวรรณ์

นาย นพดล สุวรรณเฉษฎา

..... อาจารย์ที่ปรึกษา

(ผศ.ธนิตย์ ตรีสุวรรณวัฒน์)

..... อาจารย์ที่ปรึกษา

(อ.ประสิทธิ์ จุลเสรีวงศ์)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

โครงงานนี้ สามารถสำเร็จและลุล่วงได้ต้องขอขอบพระคุณ อาจารย์ ประสิทธิ์ จุลเสรีวงศ์ และ ผศ. ธนิตย์ ตรีสุวรรณวัฒน์ ที่ได้เปิดโครงงานนี้ขึ้น และยังให้คำแนะนำในการแก้ปัญหาต่างๆ ที่เกิดขึ้นระหว่างทำโครงงาน รวมถึงท่านอาจารย์ต่างๆ เช่น อาจารย์ วิริยะ กองรัตน์ , ผศ. วิทยา ทิพสุวรรณพร , อาจารย์ ประภาษ อุคคกิมาพันธุ์ , อาจารย์ ทรงชัย วีระทิวีมาศ ที่ได้ช่วยเหลือในการแก้ปัญหาอย่างดี ขอขอบคุณภาควิชาเทคโนโลยีการวัดคุมทางอุตสาหกรรม และเพื่อนๆ ที่ให้กำลังใจที่ดีตลอดมา สิ่งที่จะขาดไม่ได้ คือ คุณพ่อ และ คุณแม่ ที่เป็นกำลังใจทางด้านจิตใจและทรัพย์สินด้วยดีตลอดมา อีกทั้งความรู้สึกดีๆ ที่มีต่อสถาบันแห่งนี้ จนโครงงานนี้ได้สำเร็จลุล่วงไปได้ด้วยดี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทคัดย่อ

การออกแบบระบบควบคุม เป็นการพยายามที่จะเข้าถึงระบบทางกายภาพที่เรียกว่า Plant หรือ Process โดย Plant จะถูกควบคุมจากตัวควบคุมซึ่งทำหน้าที่ผลิตสัญญาณควบคุมให้อย่างเหมาะสม ดังนั้นการทำงานของระบบควบคุมนั้นจำเป็นที่จะต้องถูกกำหนดโดยการออกแบบตัวควบคุมและวิธีการควบคุมที่เหมาะสม

ในสมัยก่อนตัวควบคุมยังเป็นอนาล็อกอยู่ เพราะว่าถูกสร้างขึ้นด้วยอุปกรณ์ที่เป็นอนาล็อก เช่น ออปแอมป์ อย่างไรก็ตามในปัจจุบันนี้การพัฒนาทางด้านไมโครโปรเซสเซอร์ ทำให้เป็นไปได้ที่จะสร้างตัวควบคุมโดยใช้ไมโครโปรเซสเซอร์ที่มีขนาดเล็ก สามารถหาได้ง่าย และราคาไม่แพง ซึ่งในปัจจุบันนี้ตัวควบคุมส่วนที่ใช้ในอุตสาหกรรมได้เปลี่ยนมาใช้ตัวควบคุมที่เป็นแบบดิจิตอล

ทฤษฎีการควบคุมแบบดิจิตอลมีการพัฒนาไปอย่างรวดเร็วควบคู่ไปกับวิธีการควบคุมระบบ ให้มีความสามารถมากกว่าแต่ก่อน บางส่วนถูกนำไปประยุกต์ใช้กับระบบอุตสาหกรรมและพิสูจน์ได้ว่าให้ผลที่ดี

โครงงานนี้เป็นเพียงการเลียนแบบการควบคุมกระบวนการ ที่เรียกว่าระบบ Distributed Control System (DCS) ซึ่งเป็นเพียงขั้นตอนของการเริ่มในการพัฒนาวิธีการควบคุมดังกล่าว เพื่อนำไปสู่การพัฒนาวิธีการควบคุมที่ดีกว่า ในแง่ของความรวดเร็วในการทำงาน , ประสิทธิภาพในการทำงาน และความสะดวกในการใช้งานเป็นต้น

ส่วนประกอบของโครงงานนี้แบ่งเป็น 7 ส่วน คือ ส่วนของดิจิตอลอินพุท , ดิจิตอลเอาท์พุท , อนาล็อกอินพุท , อนาล็อกเอาท์พุท , ตัวควบคุมแบบพีไอดี , ตัวควบคุมแบบอนาล็อกอินพุทเอาท์พุท และส่วนของตัวเปลี่ยนสัญญาณมาตรฐาน RS-232 เป็น RS-485 โดยการนำเอาไมโครคอมพิวเตอร์เข้ามาประยุกต์ใช้ในการควบคุมกระบวนการ ซึ่งไมโครคอมพิวเตอร์นี้จะทำหน้าที่เป็นตัวสั่งการและเป็นตัวเฝ้าคุมระบบ โดยจะติดต่อสื่อสารกับตัวไมโครคอนโทรลเลอร์ในจุดต่างๆ ด้วยสายสัญญาณมาตรฐาน RS-232 ซึ่งจะเปลี่ยนเป็นสายสัญญาณมาตรฐาน RS-485 เพื่อเข้าไปในส่วนของไมโครคอนโทรลเลอร์ ประโยชน์ของสายสัญญาณ RS-485 คือเป็นสายที่สามารถยาวได้ถึง 4000 ฟุต ซึ่งเป็นที่นิยมใช้มากในวงการอุตสาหกรรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

Abstract

A control system design is an attempt to manipulate a physical system called plant or process , in such a way that it behaves in a desired manner . The plant is contrlled by a controller which produces a suitable control signal . Hence , the performance of the conrol system is essentially determined by a proper design of the controller and the control method.

In the past , the controllers were analog because they were built from analog components , such as operational amplifiers , pneumatic amplifiers . Nowadays , however , the development of microprocessor technology makes it possible to implement the controllers using microprocessors which are compact and easily available at low cost . Most of the controllers currently employed in industrial processes are digital controllers

Digital control theories and control method advance very rapidry during the past years . Some of them have been applied in the industry and proved to be better than .

This project is a simulator which can control process . It is called "Distributed Control System" (DCS) . It is the basic development of the control method to be better than , such as speed of processing , efficiancy of processing and comfort for application .

This project combines 7 parts as Digital input unit , Digital output unit , Analog input unit , Analog output unit , PID unit , Analog I/O unit , 232 to 485 converter . The microcomputer is applied for the control processing which is the operater and monitoring , contacting with many microcontrollers in other places by RS-232 which RS-232 is changed to RS-485 for contacting microcontrollers in other places . Benefit of RS-485 is 4000 feet long wires . It is popular thing for using in industrial factories .

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

หน้า

บทที่

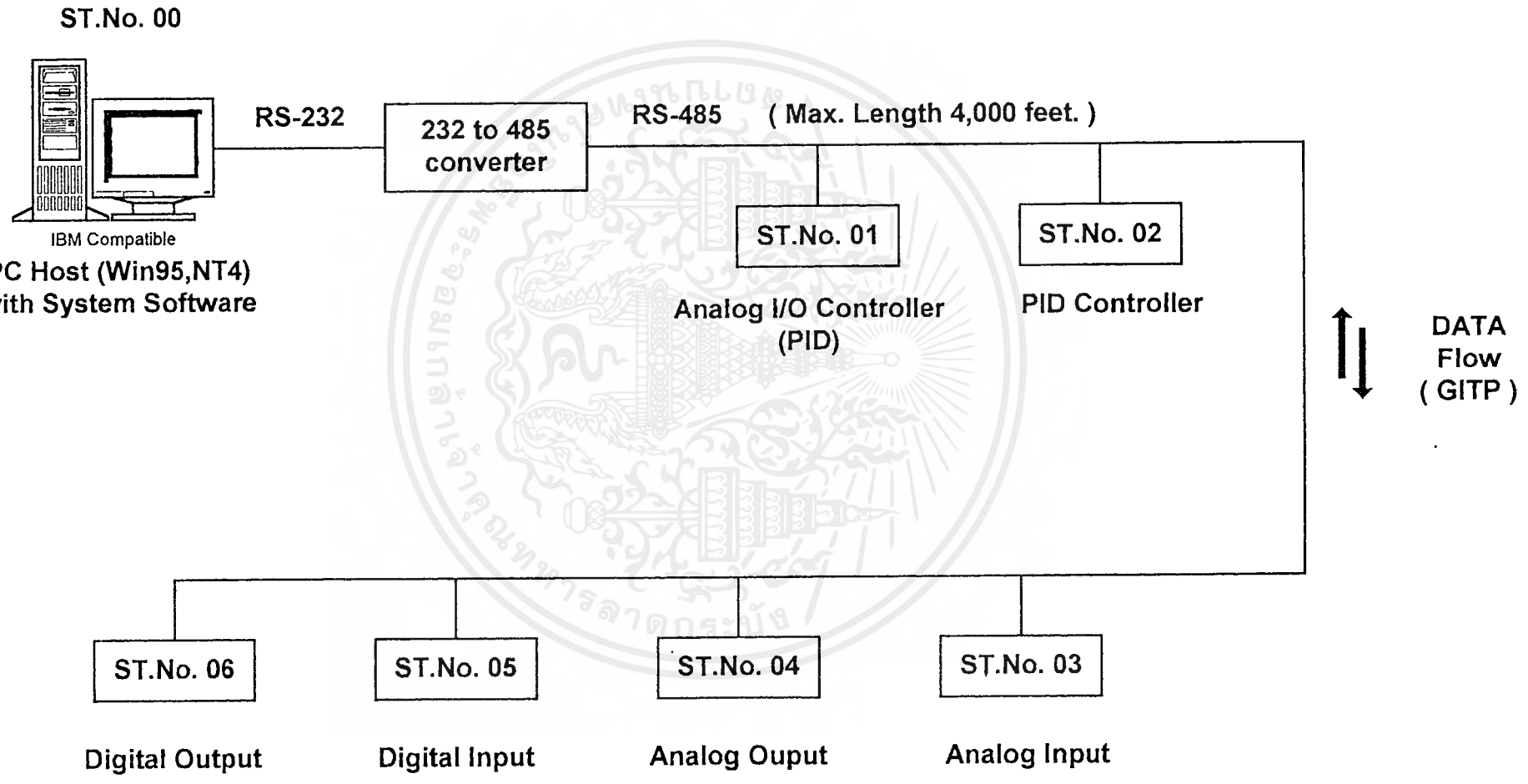
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Small Scale DCS



บทที่ 1

บทนำ

1.1 บทนำ

สมัยก่อนการควบคุมกระบวนการต่างๆ นั้นจะใช้การควบคุมอุปกรณ์เครื่องมือวัดต่างๆ ด้วยมนุษย์ โดยจะควบคุมที่บริเวณพื้นที่ของกระบวนการตามจุดต่างๆ ในโรงงาน และต่อมาได้นำเอาตัวควบคุมซึ่งเป็นแบบอนาล็อกมาใช้ในการควบคุมกระบวนการแทนที่มนุษย์ ซึ่งมีข้อจำกัดและข้อยุ่งยากพอสมควร และที่สำคัญไม่สะดวกในการปรับแต่งค่าต่างๆ นัก ทำให้มีผู้พยายามคิดและสร้างที่จะทำให้ตัวควบคุมและการควบคุมกระบวนการต่างๆ นั้นมีประสิทธิภาพเพิ่มขึ้น รวมถึงความสะดวกและรวดเร็วในการใช้งานด้วย จึงทำให้ไมโครโปรเซสเซอร์และไมโครคอมพิวเตอร์ เข้ามามีบทบาทในการควบคุมมากขึ้น โดยเฉพาะคุณสมบัติข้อดีต่างๆ ของไมโครโปรเซสเซอร์และไมโครคอมพิวเตอร์ ทำให้มีผู้นิยมกว้างขวางมากขึ้นตามลำดับ ไม่ว่าจะเป็นความรวดเร็วในการทำงาน, ความยุ่งยากในการเชื่อมต่อระหว่างอุปกรณ์ต่างๆ ที่ลดลง และความถูกต้องแม่นยำ จากคุณสมบัติเหล่านี้จึงทำให้การควบคุมสมัยใหม่เป็นระบบระเบียบ และประสิทธิภาพการทำงานสูงขึ้นกว่าแต่ก่อนมาก

จากเหตุผลและข้อจำกัดต่างๆ ของวิธีการควบคุมกระบวนการในสมัยก่อน ทำให้เกิดแนวคิดที่จะสร้างตัวควบคุมเอนกประสงค์ที่ใช้ไมโครคอนโทรลเลอร์ และใช้คอมพิวเตอร์ เป็นตัวตัวสั่งการ ซึ่งคอมพิวเตอร์นี้จะทำหน้าที่ในการติดต่อระหว่างไมโครคอนโทรลเลอร์ในจุดต่างๆ เพื่อให้สามารถทำงานได้ตามที่ต้องการ

โครงงานนี้เน้นที่จะสร้างตัวควบคุมในลักษณะต่างๆ โดยใช้คอมพิวเตอร์เป็นตัวควบคุม ดูแลตัวควบคุมในจุดต่างๆ ซึ่งจะทำให้ข้อยุ่งยากลดลง แต่ประสิทธิภาพในการทำงานสูงขึ้น ไม่ว่าจะเป็นการปรับแต่งค่าพารามิเตอร์, การแสดงผลของอินพุท เอาท์พุท และการประมวลผลต่างๆ จะเป็นไปอย่างรวดเร็ว เพราะการสั่งงาน และแสดงผลจะทำได้โดยใช้คอมพิวเตอร์เป็นตัวทำงาน

1.2 วัตถุประสงค์ของงานวิจัย

ในโครงงานนี้เป็นเพียงพื้นฐานของการพัฒนาวิธีการควบคุมกระบวนการ วิธีการดังกล่าวเราสามารถเรียกว่า Distributed Control System (DCS) ซึ่งนิยมใช้กันอยู่โดยทั่วไปในงานอุตสาหกรรม โดยพยามยามใช้ข้อดีต่างๆ ของมาตรฐานการสื่อสารอนุกรม RS-485 มาใช้ปฏิบัติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และพัฒนาในการสื่อสารระหว่างไมโครคอมพิวเตอร์กับไมโครคอนโทรลเลอร์ในจุดต่างๆ ที่สำคัญได้ทำเป็นต้นแบบเพื่อไปทดลองใช้งานกับกระบวนการได้

1.3 ขอบเขตของงานวิจัย

จากการที่มีการพัฒนาของไมโครโปรเซสเซอร์และไมโครคอมพิวเตอร์ในด้านต่างๆ เช่น ความรวดเร็วในการประมวลผล, การคำนวณทางคณิตศาสตร์, ความจุของหน่วยความจำ และความง่ายต่อการเปลี่ยนแปลงหรือเพิ่มเติมของโปรแกรม เป็นต้น จึงทำให้มีความสนใจที่จะนำไมโครโปรเซสเซอร์ และไมโครคอมพิวเตอร์มาประยุกต์ใช้ในงานควบคุมต่างๆ

โดยในโครงงานนี้จะประกอบด้วย 7 ส่วนด้วยกันซึ่งประกอบด้วย

1.Digital Input Unit ซึ่งจะใช้ไมโครคอนโทรลเลอร์เป็นตัวควบคุมการทำงาน โดยมีไมโครคอมพิวเตอร์เป็นตัวสั่งการ มีหน้าที่ในการแปลงข้อมูลขนาด 8 บิต แบบขนาน ให้เป็นข้อมูลอนุกรม 1 บิต แล้วทำการแปลงสัญญาณข้อมูลให้เป็นสัญญาณมาตรฐานการสื่อสารอนุกรม RS-485 แล้วข้อมูลเหล่านี้จะถูกส่งไปยังคอมพิวเตอร์ทีละ 1 บิต

2.Digital Output Unit จะใช้ไมโครคอนโทรลเลอร์เป็นตัวควบคุมการทำงาน โดยมีไมโครคอมพิวเตอร์เป็นตัวสั่งการ ซึ่งจะมีหน้าที่รับข้อมูลแบบอนุกรมจากไมโครคอมพิวเตอร์ แล้วทำการแปลงเป็นข้อมูลขนาด 8 บิตแบบขนาน เพื่อนำข้อมูลนี้นำไปใช้งานต่อไป

3.Analog Input Unit จะใช้ไมโครคอนโทรลเลอร์เป็นตัวควบคุมการทำงาน โดยมีไมโครคอมพิวเตอร์เป็นตัวสั่งการ มีหน้าที่รับสัญญาณอนาล็อกจากกระบวนการ แล้วทำการส่งสัญญาณเหล่านี้ไปยังคอมพิวเตอร์ในรูปแบบของสัญญาณดิจิตอล

4.Analog Output Unit จะใช้ไมโครคอนโทรลเลอร์เป็นตัวควบคุมการทำงาน โดยมีไมโครคอมพิวเตอร์เป็นตัวสั่งการ มีห้าที่ในการรับข้อมูลแบบดิจิตอลจากคอมพิวเตอร์เข้ามา แล้วทำการแปลงเป็นสัญญาณอนาล็อก เพื่อเอาไปควบคุมกระบวนการต่างๆ ต่อไป

5.PID Unit จะใช้ไมโครคอนโทรลเลอร์เป็นตัวควบคุมการทำงาน โดยติดต่อกับคอมพิวเตอร์ผ่านทางสายสัญญาณสื่อสารอนุกรม RS-485 โดยจะรับค่าจากกระบวนการมาประมวลผลกับค่าที่ตั้งไว้ แล้วส่งค่าที่เหมาะสมออกไปควบคุมกระบวนการ ให้เป็นไปตามที่ต้องการ โดยมี Algorithm แบบสากลทั่วไป คือ ตัวแปร Kp, KI, Kd, และ Sample time ให้มีความง่ายต่อการใช้งานในรูปของสมการทั่วไป และยังสามารถแสดงผลต่างๆ ที่ LCD และคอมพิวเตอร์ได้

6.Analog I/O Controller จะใช้ไมโครคอนโทรลเลอร์เป็นตัวควบคุมการทำงาน และใช้ไมโครคอมพิวเตอร์เป็นตัวสั่งการ โดยรับค่าจากกระบวนการแล้วส่งข้อมูลไปยัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คอมพิวเตอร์เพื่อทำการประมวลผลที่คอมพิวเตอร์ แล้วส่งค่าที่เหมาะสมออกไปออกไปยังไมโครคอนโทรลเลอร์เพื่อที่จะส่งค่าออกไปควบคุมกระบวนการต่างๆ

7.232 to 485 Converter จะทำหน้าที่แปลงสัญญาณข้อมูลด้วยมาตรฐาน RS-485 จากไมโครคอนโทรลเลอร์ในจุดต่างๆ ให้เป็นมาตรฐาน RS-232 เพื่อสามารถติดต่อกับไมโครคอมพิวเตอร์ได้ ซึ่งจะเห็นได้ว่ามาตรฐาน RS-485 มีความสำคัญมาก เพราะสามารถใช้สายได้ยาวมากกว่ามาตรฐาน RS-232 ซึ่งสายสัญญาณสามารถยาวได้ถึง 4000 ฟุต เป็นที่นิยมใช้ในโรงงานอุตสาหกรรม

จะเห็นได้ว่าในแต่ละส่วนที่ได้กล่าวมาแล้วนั้น จะใช้ไมโครคอนโทรลเลอร์เป็นตัวควบคุมการทำงาน เพราะมีประโยชน์ในการสามารถพัฒนาโปรแกรมไปได้เรื่อยๆ และมีความรวดเร็วในการทำงาน เป็นต้น

1.4 ประโยชน์ที่ได้รับจากงานวิจัย

จากที่ได้กล่าวมาแล้วว่า โครงงานนี้เป็นเพียงพื้นฐานของการพัฒนาวิธีการควบคุมกระบวนการเท่านั้น ซึ่งยังไม่มีประสิทธิภาพที่ทัดเทียมกับวิธีการควบคุมกระบวนการที่ใช้กันอยู่ในปัจจุบัน แต่ก็เป็นจุดเริ่มต้นที่ดี เพื่อรอการพัฒนาระบบการควบคุมต่อไปในอนาคต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

การพัฒนาการควบคุมไปสู่ระบบการควบคุมแบบ DCS

บทนำทั่วไป

ระบบการควบคุมได้พัฒนามาเรื่อยๆ ตั้งแต่ปี พ.ศ. 2473 จนกระทั่งถึงปัจจุบันนี้ ก็ด้วยจุดประสงค์หลัก คือเพื่อตอบสนองต่อปัจจัยพื้นฐาน อันได้แก่ ความปลอดภัย, ความซับซ้อนของระบบควบคุม, ปริมาณการผลิต, คุณภาพของผลิตภัณฑ์, การแข่งขันทางการตลาด และการตอบสนองความก้าวหน้าทางด้านเทคโนโลยีที่กำลังเปลี่ยนไป อันได้แก่เทคโนโลยีทางด้านอิเล็คทรอนิคส์และคอมพิวเตอร์, เทคโนโลยีทางด้านการสื่อสาร ตลอดจนเทคโนโลยีทางด้านวัสดุเป็นต้น โดยเฉพาะเทคโนโลยีทางด้านคอมพิวเตอรมีการประมาณกันว่าทุกๆ 10ปี ความสามารถของคอมพิวเตอร์จะเพิ่มขึ้นเป็น 10 เท่า ทั้งในแง่ความจุและความเร็วทางด้านการประมวลผล ดังมีคำกล่าวกันว่า " TO INCREASE TEN TIMES EVERY TEN YEARS " ดังนั้นเพื่อที่จะตอบสนองต่อเหตุผลที่กล่าวมา ทางบริษัทผู้ผลิตอุปกรณ์เครื่องมือวัดต่างๆ จึงได้พัฒนาระบบการควบคุมกระบวนการผลิตจากระบบการควบคุมแบบ นิวแมติกส์(ระบบลม) , ระบบการควบคุมด้วย analog electronics controller ไปสู่ระบบการควบคุมด้วยคอมพิวเตอร์ตามลำดับ ครั้งแรกได้นำระบบควบคุมด้วยคอมพิวเตอร์มาใช้เฉพาะงานเฝ้าคุม (MONITORING) ขบวนการผลิต แล้วจึงพัฒนาไปสู่ระบบ Supervisory Process Control (หรือเรียกอีกอย่างว่า Set point control) เรื่อยไปถึง Direct Digital Control (DDC) จนกระทั่งกลายมาเป็นระบบการควบคุมแบบ Distributed Control System (DCS) ดังเช่นในปัจจุบัน

แนวความคิดทั่วไปของ process และการควบคุม

2.1.แนวความคิดทั่วไปของ process

ความหมายของ process ในจุดประสงค์ของงานวิศวกรรมวัดคุมจะหมายถึง ขบวนการทางกายภาพที่เราต้องการควบคุมให้มีสภาวะตามต้องการ แม้ว่าสภาพแวดล้อมอาจเปลี่ยนแปลง หรือได้รับสิ่งรบกวนจากภายนอกก็ตาม

แบบจำลองของ process ดังรูปที่ 2.1 ประกอบด้วย

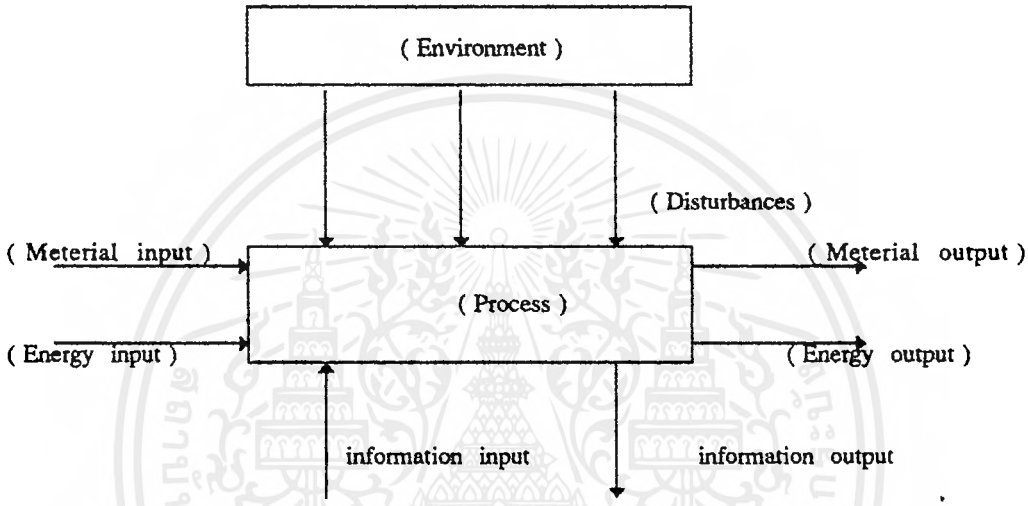
1.วัสดุ (Materials)

2.พลังงาน (Energy)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.สารสนเทศ (Information) อันได้แก่ ข้อมูลของตัวแปรที่ต้องการควบคุม, คำสั่งในการควบคุมรวมทั้งความรู้เฉพาะด้านที่ใช้ในขบวนการผลิต

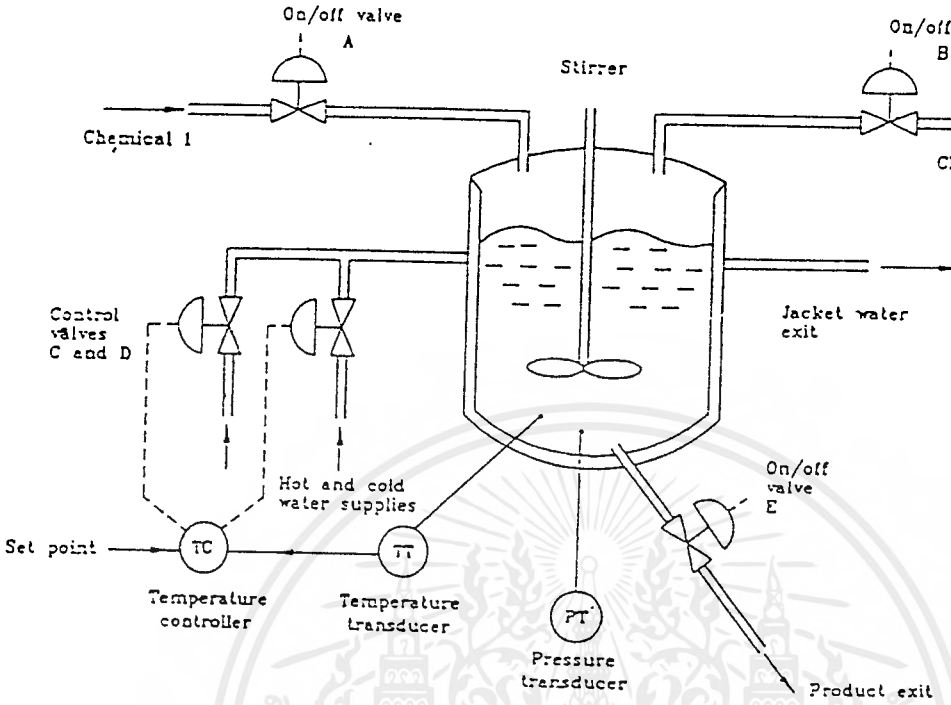
4.สิ่งรบกวน (Disturbances)



รูป 2.1 แบบจำลองของ process

ตัวอย่าง การควบคุมอุณหภูมิภายใน chemical Reactor ดังรูป 2.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 2.2

Process : Chemical Reaction

วัสดุขาเข้า : สารเคมี หมายเลข 1 และ 2 ที่ผ่านวาล์วปิดเปิด A,B ตามลำดับ

วัสดุขาออก : ผลิตภัณฑ์(product)ที่ผ่านวาล์วปิดเปิด E

พลังงานขาเข้า : พลังงานความร้อนจากน้ำร้อน (Hot Water) พลังงานกลของใบพัดกวน (Stirrer)

พลังงานขาออก : พลังงานความร้อนจากปฏิกริยาการคายความร้อน

สารสนเทศขาเข้า : สัญญาณที่บอกถึงเปอร์เซนต์ในการเปิดวาล์วของน้ำร้อนและน้ำเย็น

สารสนเทศขาออก : อุณหภูมิภายใน reactor

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

: ความดันภายใน reactor

สิ่งรบกวน : การเปลี่ยนแปลงอุณหภูมิของน้ำร้อนและน้ำเย็นที่จ่ายให้แก่ reactor

: การสูญเสียพลังงานความร้อนที่ท่อและผนังของ jacket

: การเปลี่ยนแปลงการจ่ายของสารเคมี หมายเลข 1,2

: การเปลี่ยนแปลงปริมาณความต้องการของผลิตภัณฑ์

2.2.แนวความคิดของการควบคุม (concept of control)

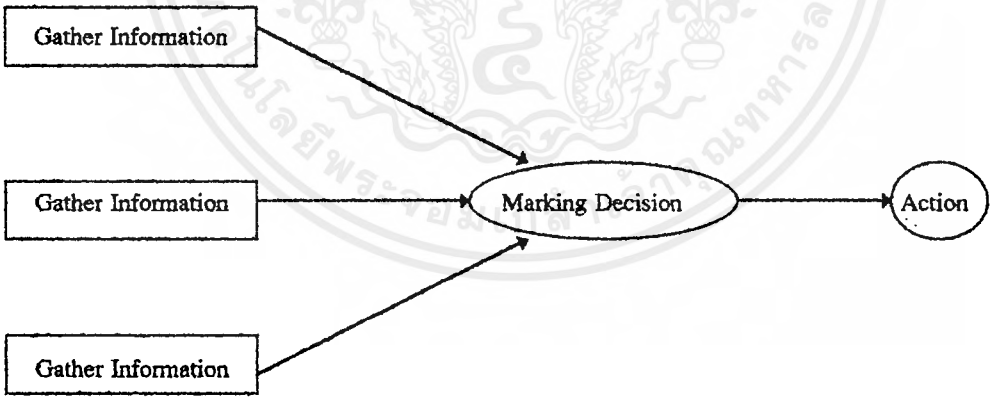
2.2.1 การควบคุมกระบวนการ หมายถึง การดูแลกำกับส่วนต่างๆ ของ process ให้เป็นไปตามความต้องการ ขั้นตอนหลักๆ ของการควบคุมประกอบด้วย

1.รวบรวมข้อมูลของตัวแปร หรือ พารามิเตอร์ของ process

2.นำเอาข้อมูลที่รวบรวมได้มาทำการคำนวณ, ประมวลผลและตัดสินใจ

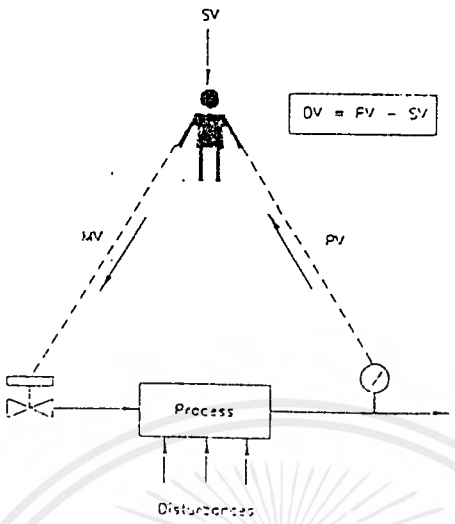
3.กระทำตามผลที่ได้จากข้อที่ 2

โดยทั้ง 3 ขั้นตอน จะทำต่อเนื่องกันไป



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.2 ตัวแปรที่เกี่ยวข้องกับการควบคุม process



รูปที่ 2.3 ตัวแปรที่เกี่ยวข้องกับการควบคุม

SV : Setpoint Variable หมายถึง ค่าเป้าหมาย

MV : Manipulated Variable หมายถึงตัวแปรปรับค่า

PV : Process Variable หมายถึง ตัวแปร process (ตัวแปรที่จะต้องถูกควบคุมเพื่อให้ได้ตามเป้าหมาย)

DV : Deviation Variable หมายถึง ตัวแปรที่แสดงค่าผิดพลาดของผลต่างระหว่าง PV กับ SP โดยมีความสัมพันธ์ถึง DV = PV - SP

2.2.3 ประเภทของวิธีการควบคุม process

วิธีในการควบคุม process พอที่จะแบ่งออกได้เป็น 2 ประเภท ดังนี้

1. ลูปควบคุม (Loop Control)

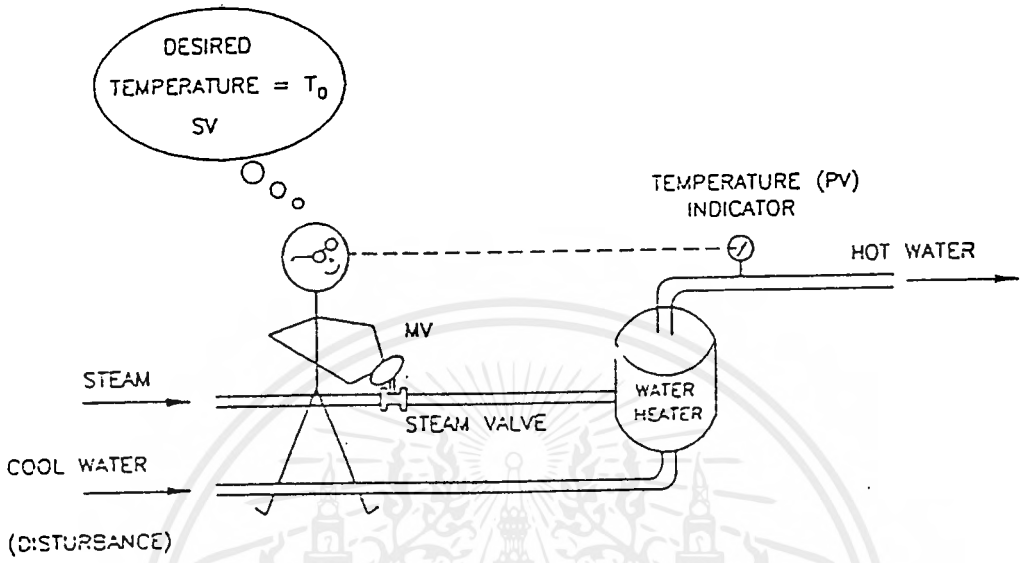
-การควบคุมแบบป้อนกลับ (Feedback Control)

-การควบคุมแบบป้อนล่วงหน้า (Feedforward Control)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. การควบคุมลำดับขั้นตอนของขบวนการ (sequence Control)

การควบคุมแบบป้อนกลับ (Feedback control)



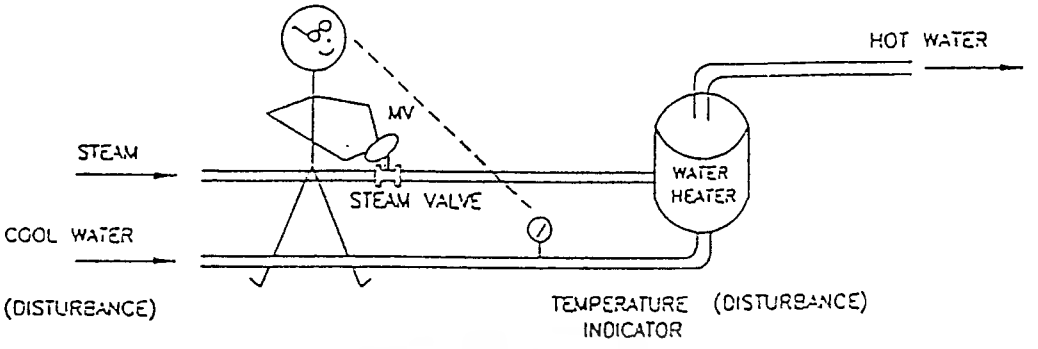
รูปที่ 2.4 ตัวอย่างการควบคุมแบบป้อนกลับ

หลักการควบคุมแบบป้อนกลับ

จากรูปที่ 2.4 ตัวแปร process(PV) ที่อ่านได้จากมิเตอร์วัดอุณหภูมิน้ำร้อนขาออกจะถูกนำไปเปรียบเทียบกับค่าอุณหภูมิเป้าหมายที่ต้องการ SV อยู่เสมอ หลังจากผ่านการประมวลผลและตัดสินใจแล้ว จึงทำการปรับเปอร์เซนการเปิดวาล์วตามสัดส่วนของค่าตัวแปรปรับค่า (MV) จะมีทิศทางที่จะรักษาค่าตัวแปร process (PV) ให้เท่ากับค่าเป้าหมาย (SV) แม้ว่าจะมีสิ่งรบกวนคอยกระทำต่อตัวแปร process เช่นการเปลี่ยนแปลงอุณหภูมิน้ำเย็นขาเข้า, อัตราการไหลของน้ำเย็น และสภาวะแวดล้อมอื่นๆ

การควบคุมแบบป้อนล่วงหน้า (Feedforward Control)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.5 ตัวอย่างการควบคุมแบบการป้อนล่วงหน้า

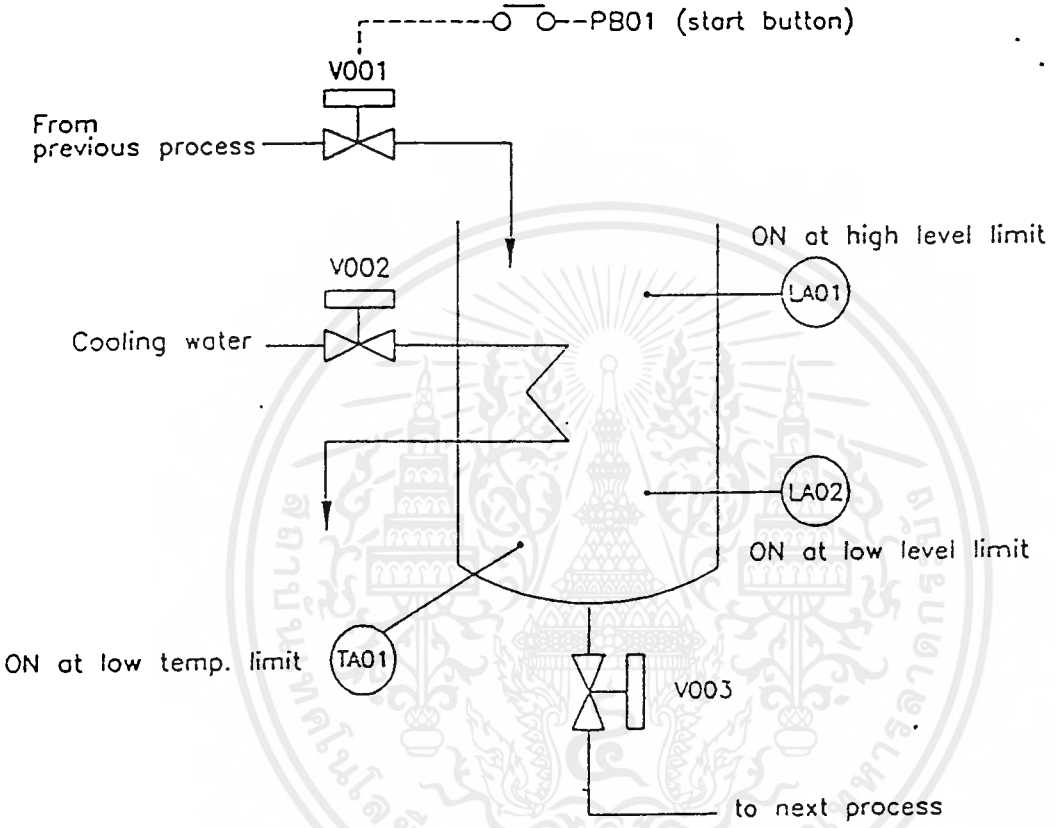
หลักการควบคุมแบบป้อนล่วงหน้า

ตัวแปรปรับ process (Manipulated Variable) ถูกกำหนดจากสิ่งรบกวน process โดยการวัดหาสิ่งรบกวน process (Process Disturbance) แล้วนำค่าที่วัดได้ไปคำนวณหาค่า MV ที่เหมาะสม " ป้อนล่วงหน้า (Feedforward) " ไปปรับ process ต้านกับสิ่รบกวนก่อนที่ตัวแปร process จะเปลี่ยนแปลงไป จากรูป 2.5 ทุกๆ ครั้งที่อุณหภูมิของน้ำเย็นซึ่งในที่นี้ คือสิ่งรบกวน เกิดการเปลี่ยนแปลงจะถูกอ่านค่าแล้วไปคำนวณผลหาค่า MV ใหม่ เปลี่ยนอัตราการไหลของไอน้ำไปชดเชยกับสิ่งรบกวนที่เกิดขึ้นก่อนที่อุณหภูมิของน้ำร้อนจะเปลี่ยนไป

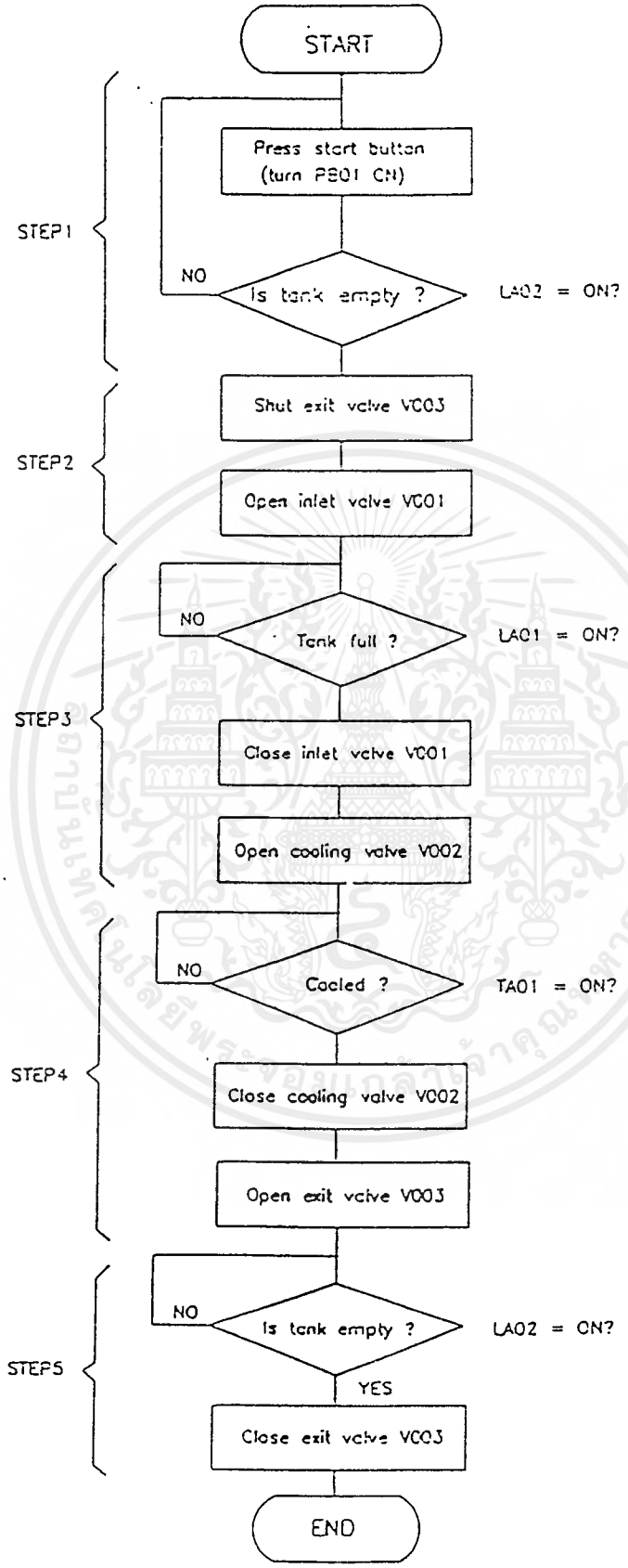
หมายเหตุ : การควบคุมแบบป้อนล่วงหน้าจะควบคุมได้เป็นอย่างดี ก็ต่อเมื่อเราทราบความสัมพันธ์ระหว่างตัวแปรปรับค่า (MV) กับสิ่งรบกวนระบบอย่างถูกต้องชัดเจน ซึ่งในทางปฏิบัติเราทราบความสัมพันธ์นี้แบบคร่าวๆ เท่านั้น เพื่อให้การควบคุมเป็นไปโดยถูกต้องจึงต้องใช้งานรวมกับระบบการควบคุมแบบป้อนกลับด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การควบคุมลำดับขั้นตอน (Sequence Control)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SYMBOL	COMMENT	RULE STEP	01	02	03	04	05	06
			1	2	3	4	5	
PB01	Start button		Y					
LA01	High level limit switch				Y			
LA02	Low level limit switch		Y				Y	
TA01	Temperature switch					Y		
V001	Inlet valve			Y	N			
V002	Cooling valve				Y	N		
V003	Exit valve			N		Y	N	
NEXT STEP	THEN		2	3	4	5	1	
	ELSE							

รูปที่ 2.6 ตัวอย่างการควบคุมลำดับขั้นตอนของ cooling process

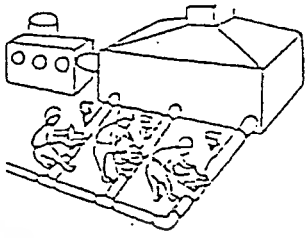
หลักการควบคุมลำดับขั้นตอน

หมายถึง การควบคุมการทำงานที่ประกอบด้วยหลายๆ ขั้นตอนย่อยให้ลำดับก่อนหลังของขั้นตอนย่อยเป็นไปตามที่เรากำหนดแน่นอนหรือเปลี่ยนแปลงตามสภาพเหตุการณ์ (Events) สภาพเหตุการณ์ดังกล่าวทราบได้จากอุปกรณ์ทางด้าน digital internal switch เป็นต้น นอกจากเป็นการเปลี่ยนแปลงลำดับขั้นตอนย่อยแล้ว ยังใช้กำหนดสถานการณ์ของการ interlock ระบบควบคุม และการ shut down ระบบ รูปที่ เป็นภาพตัวอย่างการควบคุมลำดับขั้นตอนของ cooling process

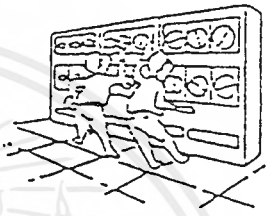
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.พัฒนาทางด้านระบบการควบคุมอัตโนมัติ

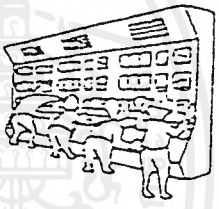
Period of 1930~
- Mechanical Instruments
- Local operations



Period of 1940~
- Large Pneumatic Instruments
- Signal transmission

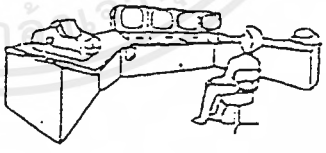


- Period of 1950~
- Small Pneumatic Instruments



Period of 1960~
- Small Electronic Instruments
- Digital Computer Control

Period of 1975~
- Distributed Control System
- CRT Operation



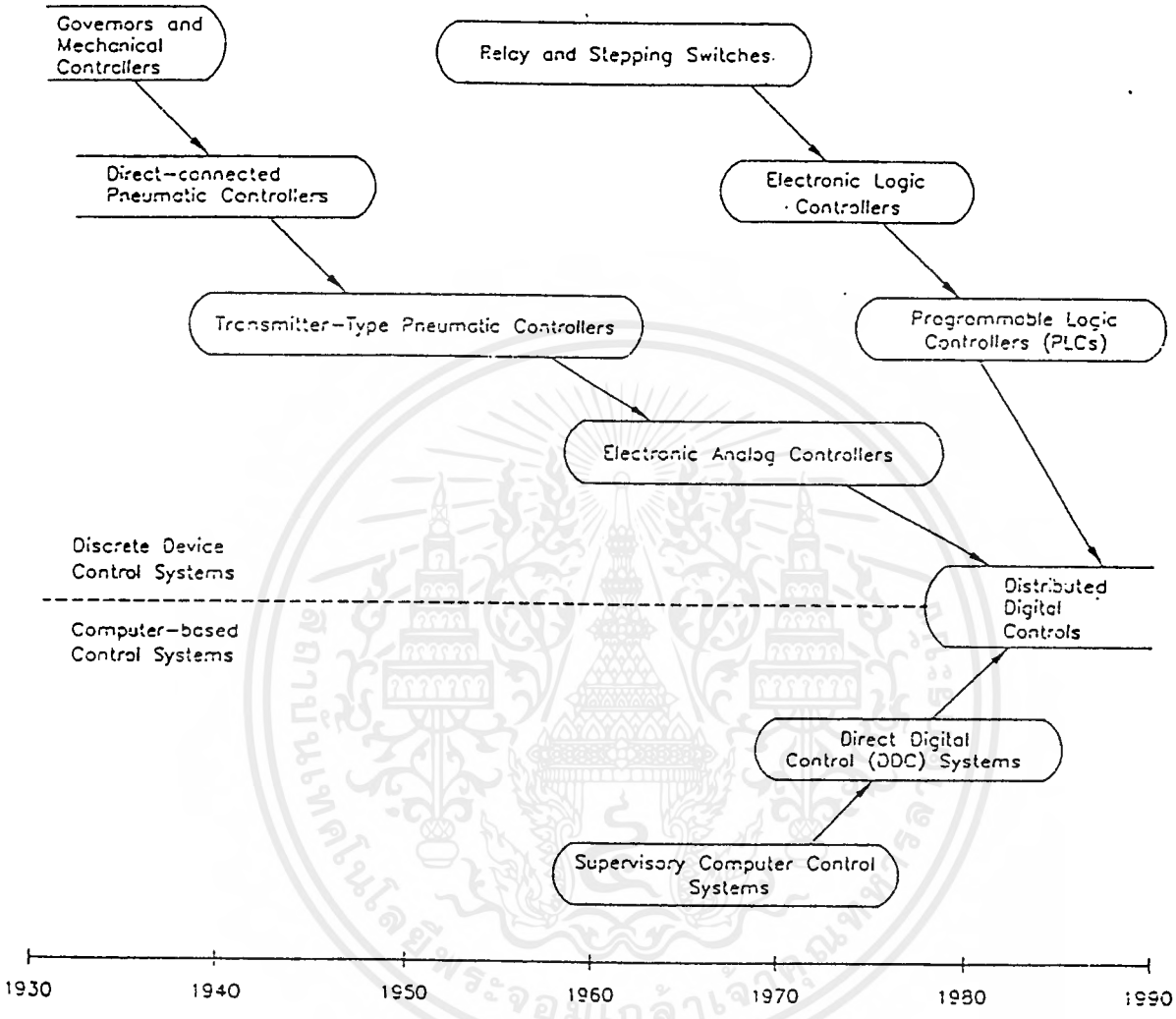
รูปที่ 2.7 วิวัฒนาการของระบบการควบคุมอัตโนมัติ

จุดเริ่มต้นของเทคโนโลยีวัดคุม

การนำเอาระบบการวัดคุมอัตโนมัติมาใช้ในงานวัดปริมาณทางด้านอุตสาหกรรม อันได้แก่ อุณหภูมิ, อัตราการไหลและความดันเริ่มเกิดขึ้นราวๆ ปีค.ศ.1930 ซึ่งใช้ในกระบวนการกลั่นน้ำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มันของบริษัท อเมริกัน ลักษณะเด่นของเทคโนโลยีวัดคุมในช่วงเวลานี้คือ เครื่องมือวัดคุมจะติดตั้งที่ห้องที่บริเวณแหล่งของขบวนการผลิตซึ่งเป็นอุปกรณ์วัดคุมที่ทำด้วยเครื่องจักรกลขนาดใหญ่ ดังรูป 2.7



รูปที่ 2.8 แนวโน้มการพัฒนาระบบการควบคุมไปสู่ระบบ Distributed Control System

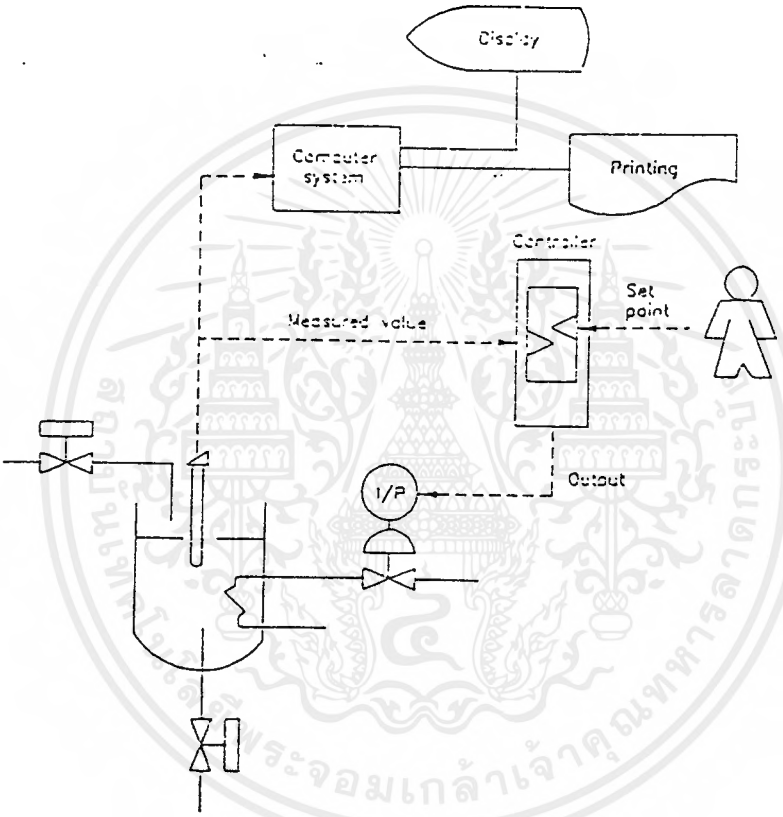
ช่วงค.ศ. 1950 ถึง 1960 คือช่วงของการผันเปลี่ยนเทคโนโลยี

ช่วงค.ศ. 1950 ถึง 1960 เทคโนโลยีทางด้านการวัดคุมได้รับการพัฒนาอย่างมาก ควบคู่ไปกับการฟื้นตัวทางด้านเศรษฐกิจของประเทศญี่ปุ่นที่มุ่งไปสู่อุตสาหกรรมเหล็ก, ปิโตรเลียม และอุตสาหกรรมสิ่งทอ ลักษณะของการวัดคุมประกอบด้วยเครื่องวัดคุมแบบ analog single loop controller จำนวนมาก โดยในช่วงเริ่มด้วยการใช้ตัวควบคุมแบบ pneumatic ที่ทำงานด้วยลม แต่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

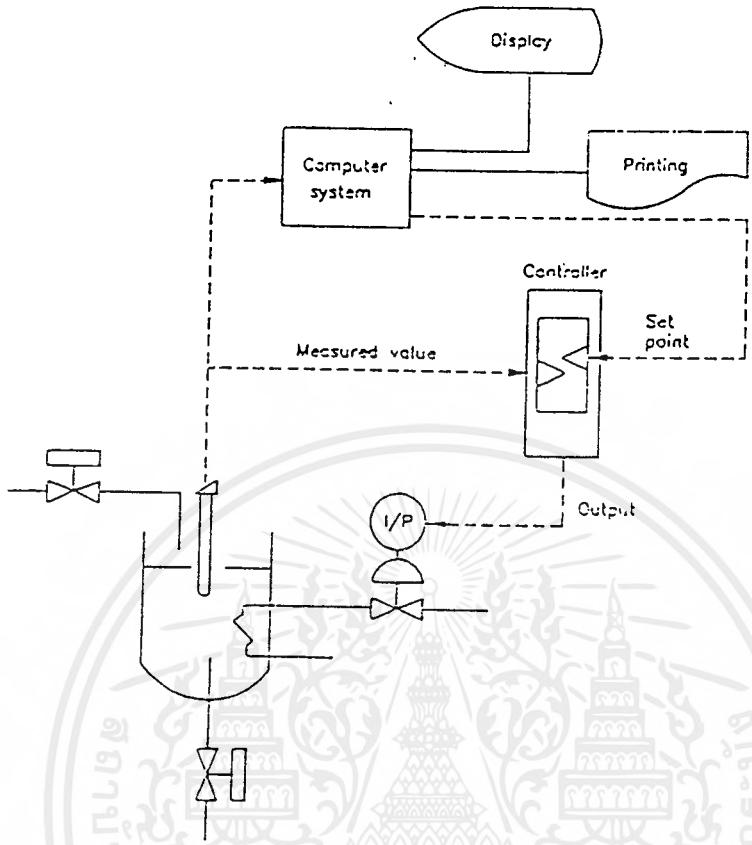
ด้วยความก้าวหน้าทางด้านอิเลคทรอนิคส์ และขนาดของกระบวนการผลิตที่ขยายใหญ่ขึ้น อุปกรณ์การวัดจึงได้พัฒนาจากระบบ pneumatic ไปสู่ระบบอิเลคทรอนิคส์มาโดยลำดับ

ค.ศ.1960 ปีแห่งจุดเริ่มต้นของคอมพิวเตอร์กับงานวัดคุม



รูปที่ 2.9 สถาปัตยกรรมการควบคุมแบบ data logging

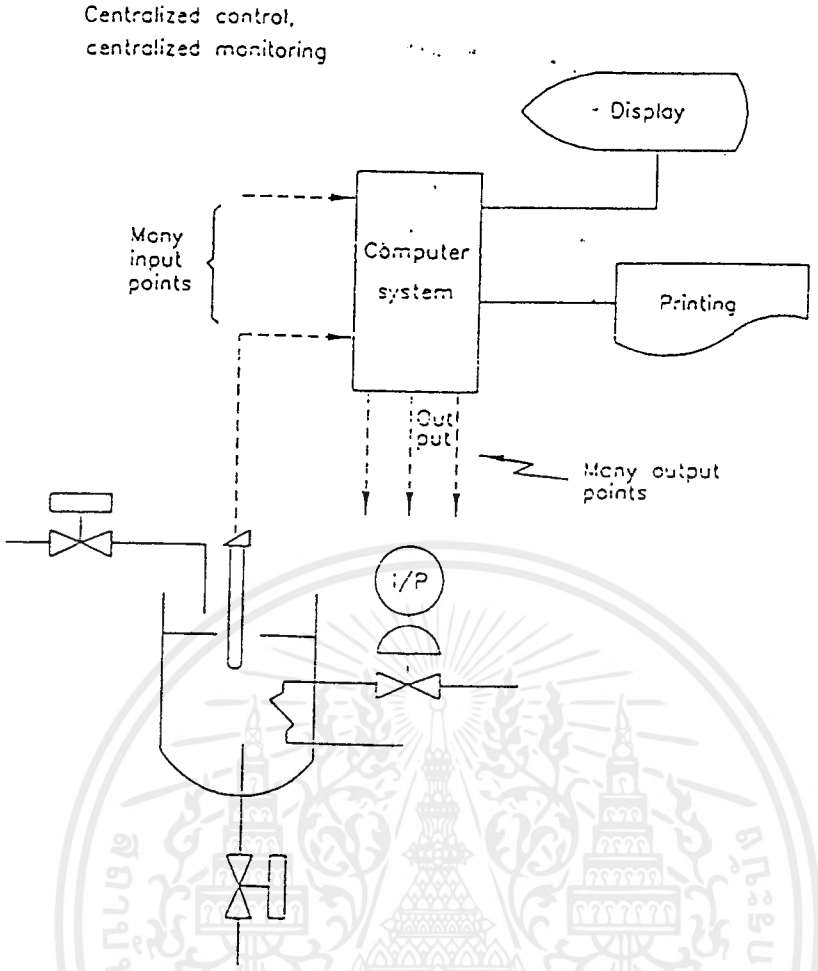
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.10 สถาปัตยกรรมการควบคุมแบบ setpoint control

ใน ค.ศ.1960 นับว่าเป็นปีแห่งจุดเริ่มต้นของการนำเอาคอมพิวเตอร์มาใช้งานควบคุมกระบวนการ โดยในครั้งแรกคอมพิวเตอร์ได้ถูกนำมาบันทึกผลและเฝ้าคุม (Recording and Monitoring) ดังรูป 2.9 ที่เรียกว่า " Data Logging " และเนื่องจากว่าคอมพิวเตอร์มีสมรรถนะทางด้านการคำนวณที่ดี จึงได้ถูกออกแบบให้ทำหน้าที่คำนวณหาเงื่อนไขที่ดีที่สุด (Optimum Process Condition) ของกระบวนการคำนวณหาค่าตัวแปร (SV) ให้แก่เครื่องควบคุมแต่ละตัว โดยสถาปัตยกรรมที่กล่าวมานี้เรียกว่า " Supervisory Process control " หรือเรียกอีกอย่างหนึ่งว่า " Set Point Control " ดังรูป 2.10

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

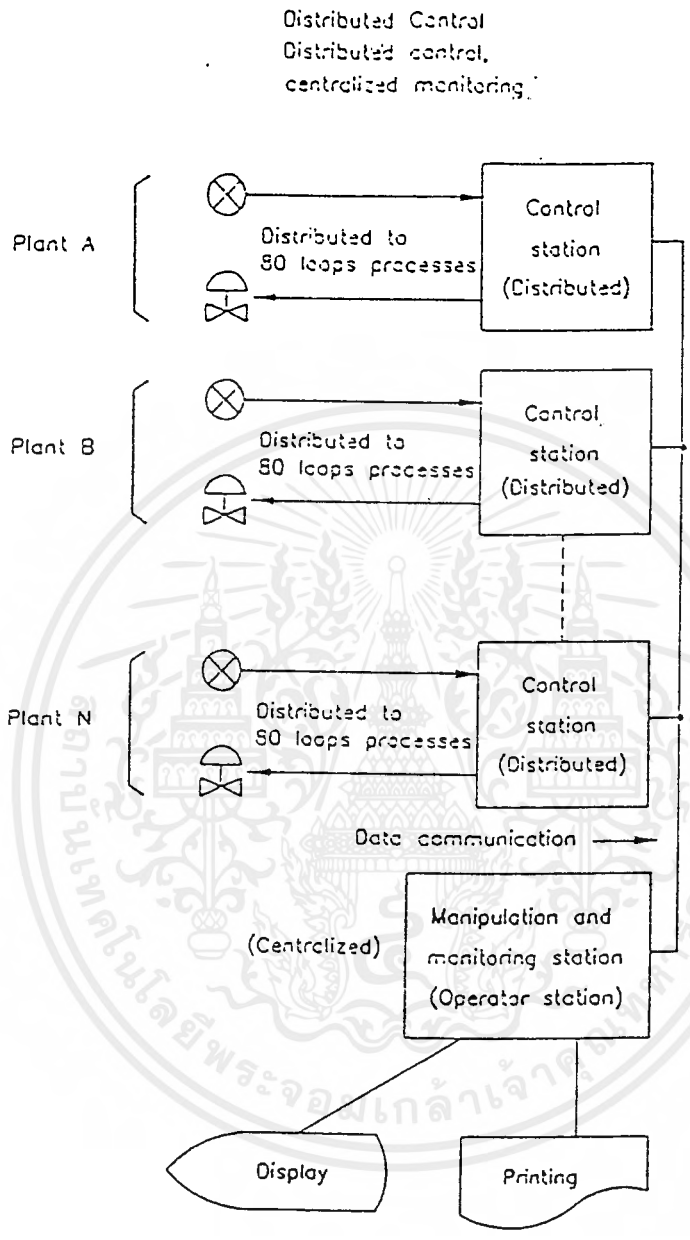


รูปที่ 2.11 สถาปัตยกรรมการควบคุมแบบ Direct Digital Control

ต่อมาประมาณปี 1965 ได้นำเอาคอมพิวเตอร์ที่มีสมรรถนะในการคำนวณสูงๆ มาใช้แทนเครื่องควบคุมแบบ analog การควบคุมนี้อาศัย digital computer เพียงหน่วยเดียวควบคุมกระบวนการแบบรวมศูนย์ และรับผิดชอบต่อ loop การควบคุมจำนวนหลายร้อย loop สถาปัตยกรรมที่กล่าวมาทั้งหมดนี้มีชื่อเรียกว่า " Direct Digital Control " หรือเรียกย่อๆ ว่า " DDC " แต่ปัญหาที่ตามมาอันเนื่องจากการล้มเหลวของคอมพิวเตอร์ที่คาดไม่ถึง ได้ส่งผลทำให้การปฏิบัติการต่อกระบวนการผลิตได้หยุดชะงักงัน (Plant Operation to a Halt) ดังนั้นการออกแบบด้วยการสำรองอุปกรณ์ดังเช่น CPU, อุปกรณ์การ back up และอื่นๆ จึงกลายเป็นสิ่งจำเป็น แต่เมื่อคำนึงถึงค่าใช้จ่ายที่จำเป็นต้องมีสำหรับการสำรองอุปกรณ์แล้วมีผลทำให้ระบบ DDC ไม่สามารถแทนที่ระบบการควบคุมแบบ analog ได้ทั้งหมด เพราะว่ามีราคาแพงมากในตอนนั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ค.ศ.1975 ปีแห่งการเริ่มต้นของ DCS

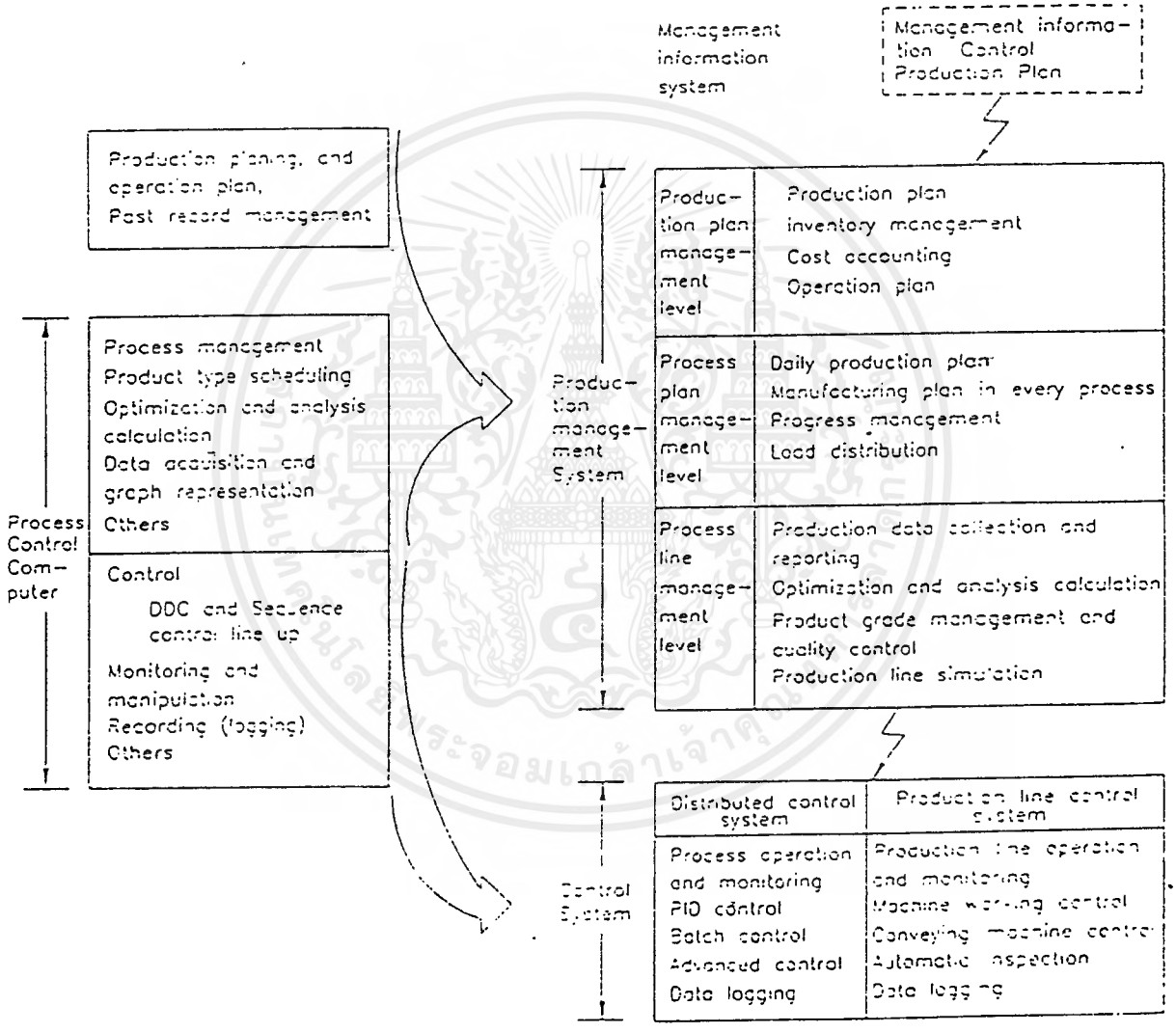


รูปที่ 2.12 สถาปัตยกรรมการควบคุมแบบ Distributed Control System

การกำเนิดของ microprocessor ในปี ค.ศ.1970 ได้นำไปสู่การเปลี่ยนแปลงในหลายสาขา รวมทั้งทางด้านการวิศวกรรมวัดคุมด้วยราคาของ microprocessor ที่ลดลง กอปรกับสมรรถภาพที่เพิ่มสูงขึ้นจึงเป็นผลให้ระบบควบคุมที่ใช้เพียงระบบควบคุมด้วยคอมพิวเตอร์เพียง 1 หน่วย ควมคุมแบบรวมศูนย์ (เพราะติดปัญหาในด้านราคาของ processor) สามารถที่จะเป็นแบบระบบการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ควบคุมแบบกระจาย หรือเรียกว่า "Distributed Control System" ระบบนี้มี chip microprocessor กระจายไปอยู่ตามแต่ละ control station ของแต่ละพื้นที่ของกระบวนการผลิต microprocessor แต่ละตัวที่กระจายอยู่ตามแต่ละ station สามารถสื่อสารกันกับ operator station ทาง terminal ของจอภาพเพื่อใช้ในการเฝ้าคุมและปฏิบัติการ หลังจากปีค.ศ. 1975 บรรดาผู้ผลิตทั่วโลกได้เริ่มขยายการประยุกต์ต่อการควบคุมแบบ batch และควบคุมลำดับขั้น (Batch and Sequence Control)



รูปที่2.13บทบาทที่เปลี่ยนของคอมพิวเตอร์ที่ใช้ควบคุมกระบวนการผลิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นอกจากนี้ ในปี ค.ศ.1970 ทางสมาคม IEC (International Electrotechnical Commission) ได้ประชุมร่างมาตรฐานเกี่ยวกับการส่งสัญญาณกระแสไฟฟ้าขนาด 4 - 20 mA DC เพื่อใช้ในการเชื่อมต่อเครื่องวัดคุมทางอุตสาหกรรมแบบ analog ที่ต่างผู้ผลิต และต่อมาเมื่ออุปกรณ์วัดคุมอุตสาหกรรมแบบใหม่ได้ได้รับการพัฒนาไปสู่ระบบ digital การเชื่อมต่อ (Interface) ระหว่างอุปกรณ์ (Devices) เปลี่ยนจากการส่งสัญญาณเป็น analog เป็นการส่งข้อมูลข่าวสารจำนวนมาก (Information) แบบ digital มาตรฐานการสื่อสารข้อมูลจึงเริ่มมีความสำคัญมากขึ้นเรื่อยๆ และด้วยการคำนึงถึงเหตุผลในข้อนี้ ทางด้านสมาคม IEC จึงได้ร่างระบบ BUS มาตรฐานสำหรับเครื่อง DCS ที่เรียกว่า PROWAY แต่อย่างไรก็ตามในสถานการณ์ของอุตสาหกรรมตอนนี้ยังคงมีระบบ BUS อิสระแตกต่างกันไปตามแต่ผู้ผลิต ในปัจจุบันนี้ เทคโนโลยีทางด้านวิศวกรรมคอมพิวเตอร์และการสื่อสารข้อมูลได้เจริญกาวหน้าไปอย่างรวดเร็ว คอมพิวเตอร์ได้รับการขยายบทบาทจากงานทางด้านการควบคุมไปสู่งานทางด้านบริหารทางด้านการผลิต (Production Management System) และระบบการบริหารงานสารสนเทศ (Management Information System) ดังรูป 2.13

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.ข้อเปรียบเทียบทางสถาปัตยกรรมของระบบควบคุมด้วยคอมพิวเตอร์ชนิดต่างๆ

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

ตารางที่ 2.1 ตารางเปรียบเทียบระบบควบคุมด้วยคอมพิวเตอร์ชนิดต่างๆ

038016

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.ข้อเปรียบเทียบสารสนเทศในแต่ละระดับ

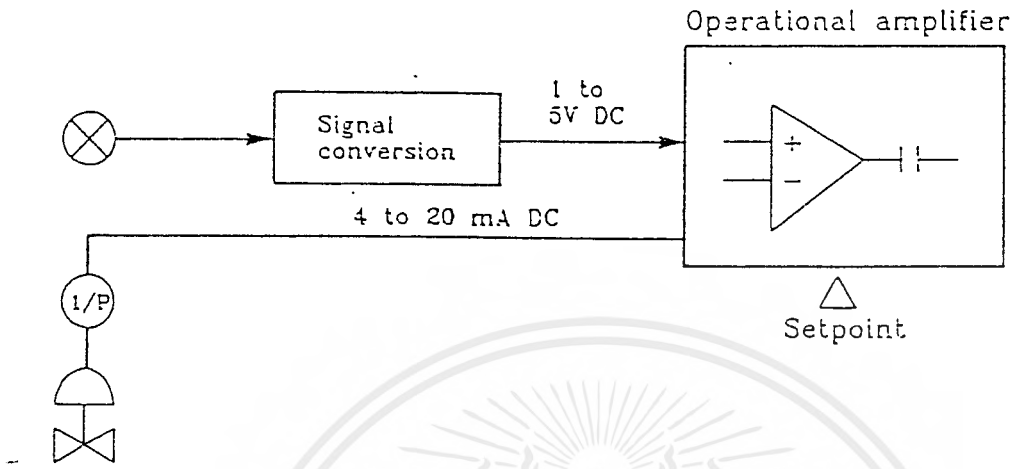
LEVEL	REALM	PLANNING PERIOD	MANAGEMENT UNIT	DATA ACQUISITION CYCLE	RESPONSE TIME (TRANSACTION PROCESSING)	DATA RETENTION TIME	DATA BASE CAPACITY
Management information system	Entire company	Months to a few years	Months	Months	Several tens minutes	1 year or more	Several hundreds M bytes or more
Production planning (management level) Production planning Inventory control Cost calculation Order control etc	Factory	1 month to 6 months	Weeks or months	Days	Several tens seconds	6 months to 1 year	Several hundreds M bytes or more
Schedule control (management level) Production plan for each process Progress control load distribution etc	Production dept	Weeks to months	Per shift or day	Hours	A few seconds	1 month to 3 months	Several tens M bytes to several hundred. M bytes
Process line management Monitoring Data acquisition Analysis calculation etc	Production section	For each shift and day	Minutes or hours	Minutes	2 sec. or less	Days to weeks	Several M bytes to several tens M bytes
Operation Monitoring Data acquisition [Continuous process] Control Alarm [Discrete process] Control Alarm	Production line process	(Real time)		(Real time) Scan cycle 1 to 10 sec Scan cycle 1 ms to 10 sec	1 sec or less 100 ms or less 100ms to 10 ms	A few ms to 1 hour	Several tens K bytes to several hundreds K bytes

ตาราง2.2 ตารางเปรียบเทียบสารสนเทศในแต่ละระดับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6.ข้อเปรียบเทียบทางด้าน hardware ของระบบการควบคุมแบบ analog และควบคุมแบบ digiltal

2.6.1.ระบบการควบคุมแบบ analog



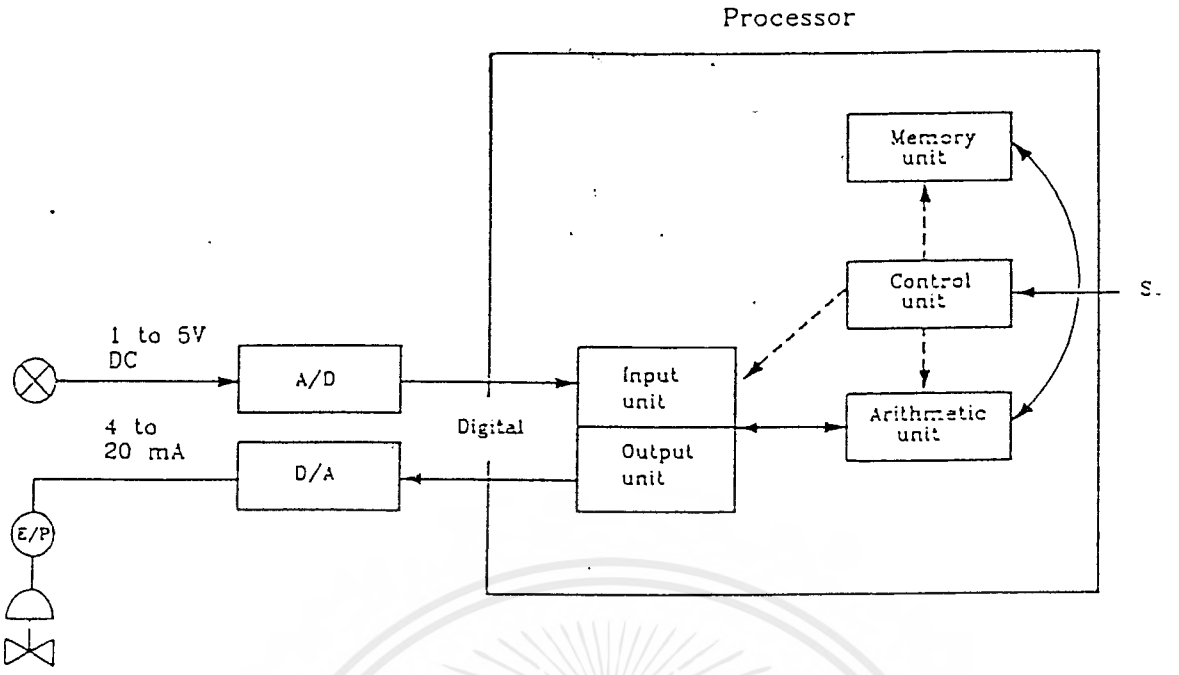
รูปที่ 2.14 Hardware โดยทั่วไปของระบบควบคุมแบบอนาล็อก

ระบบการควบคุมแบบ analog (Analog Control System) : หน่วยควบคุมที่ประมวลผลการควบคุมกับสัญญาณ analog (เช่น แรงดันไฟฟ้า) ด้วยอุปกรณ์ประเภทตัวขยายสัญญาณ (Operational Amplifier) โดยการควบคุมแบบลำดับกระบวนการ (Sequence Control) ยังไม่มีในที่นี้

2.6.2. ระบบการควบคุมแบบ digital

ระบบการควบคุมแบบ Digital (Digital Control System) : หน่วยควบคุมที่ประมวลผลการควบคุมกับข้อมูล digital ด้วยอุปกรณ์ประเภทโปรเซสเซอร์ (Processing unit) ระบบนี้มีการควบคุมทั้งการควบคุมแบบป้อนกลับ (Feedback Control) และการควบคุมป้อนล่วงหน้า (Feedforward Control) และการควบคุมลำดับขั้นตอนกระบวนการ (Sequence Control) เนื่องจากการประมวลผลเป็นแบบ digital ด้วยเลขฐาน 2 จึงจำเป็นต้องมีอุปกรณ์ในการแปลงสัญญาณ analog เป็น digital เรียกว่า A/ D (Analog to Digital Convertor) หลังจากนั้นจึงส่งไปยังหน่วย input (Input unit) ของโปรเซสเซอร์ รวมทั้งอุปกรณ์แปลงสัญญาณจาก digital เป็น analog ที่รับผลการประมวลจากหน่วย output (Output unit) ที่เรียกว่า D/A (Digital to Analog Convertor)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

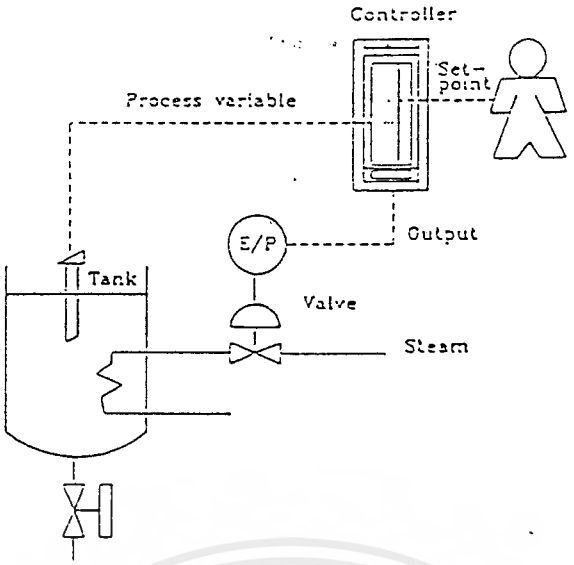


รูปที่ 2.15

7.การเปรียบเทียบการควบคุมแบบป้อนกลับระหว่าง DCS กับ analog controller

รูป 2.16 แสดงถึงการควบคุมอุณหภูมิภายใน reactor ตามวิธีการควบคุมแบบป้อนกลับ (Feedback Control) ด้วยเครื่องควบคุมแบบ analog เจ้าหน้าที่ปฏิบัติการควบคุม (Operator) มีหน้าที่ป้อนสัญญาณเป้าหมาย (SV) ของอุณหภูมิแก่เครื่องควบคุม หลังจากนั้นเครื่องควบคุมมีหน้าที่คำนวณหาสัญญาณ MV เพื่อใช้ในการปรับวาล์วของไอน้ำ (stream) ในทิศทางที่ลดค่าผิดพลาด PV และ SV

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.16 การควบคุมโปรเซสด้วยเครื่องควบคุมแบบอนาล็อก

การคำนวณหาสัญญาณ MV หาได้จากสัญญาณของค่าที่ผิดพลาดตามขั้นตอนวิธีการควบคุมแบบ PID (PID Control Algrorithm) ดังสมการที่ (1)

$$MV(T) = \frac{100}{PB}\left[E(T) + \frac{1}{Ti}\int_0 E(t)dt + Td\frac{dE(t)}{dt} \right] \qquad (1)$$

E(t) : SV(t) - PV(t) เมื่อกำหนดเป็นแบบ Reverse Action

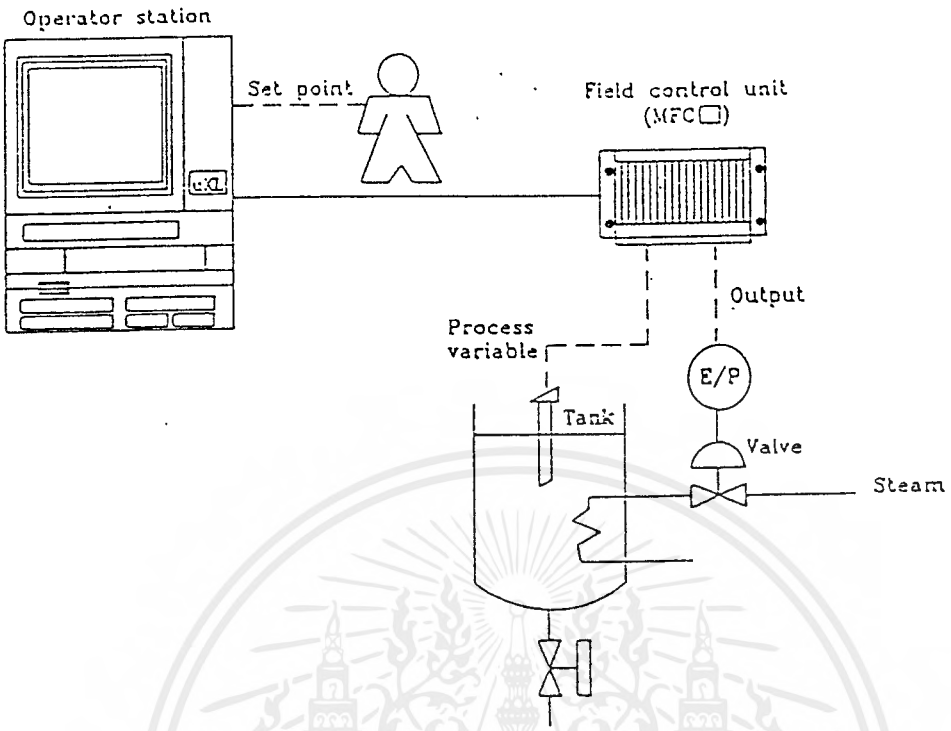
PB : P(Proportional Band) หน่วยเป็นเปอร์เซนต์ (%)

Ti : I(Integral Time) หน่วยเป็นวินาที (S)

Td : D(Derivation Time) หน่วยเป็นวินาที (S)

รูปที่ 2.17 แสดงถึงการนำเอาระบบควบคุม DCS มาใช้ควบคุมอุณหภูมิภายใน reactor โดย Operator Station ทำหน้าที่แสดงผลการควบคุมและรับคำสั่งในการปฏิบัติการ ส่วน MFCD_ Field Control Station ทำหน้าที่ควบคุมโปรเซส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

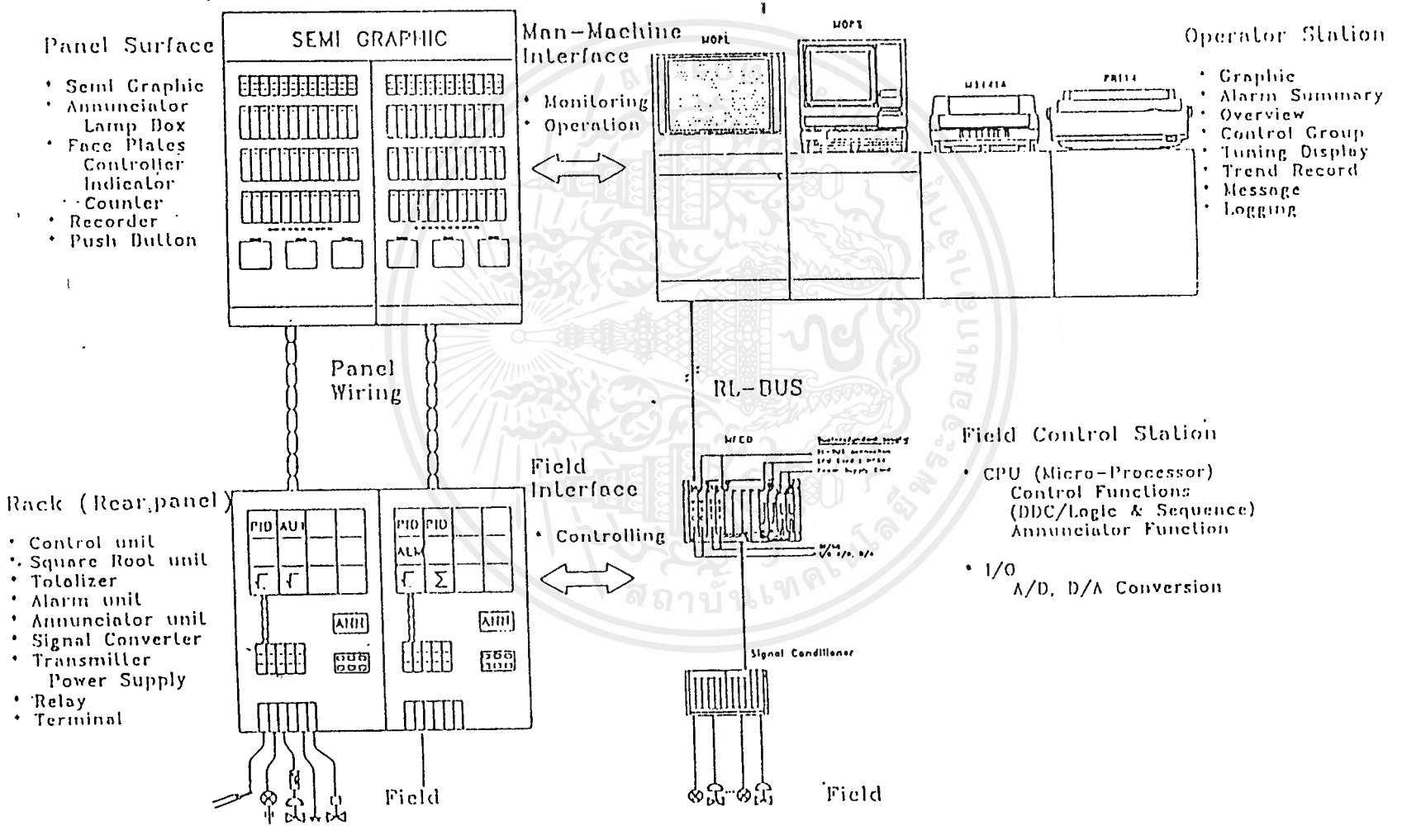


รูปที่ 2.17 Process Control by MFCD or MFCU

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CONVENTIONAL CONTROL SYSTEM

DISTRIBUTED CONTROL SYSTEM (DCS)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

สถาปัตยกรรมของไมโครคอนโทรลเลอร์แบบชิพเดี่ยวตระกูล 51

3.1 โครงสร้างของไมโครคอนโทรลเลอร์ตระกูล MCS-51

ไมโครคอนโทรลเลอร์ตระกูล MCS-51 มีสมาชิกในตระกูลหลายเบอร์ด้วยกัน แต่ละเบอร์จะมีคุณสมบัติพิเศษแตกต่างกัน เช่น มีหน่วยความจำภายในสำหรับเก็บโปรแกรมและข้อมูลภายในชิปเพิ่มขึ้น มีวงจรเปลี่ยนค่าสัญญาณอานะล็อกเป็นดิจิตอลในตัว สามารถรับสัญญาณอินเตอร์รัปต์ได้หลายชนิด ทำกระบวน DMA (**Direct Memory Access**) ได้ในตัว มีรีจิเตอร์สำหรับใช้เป็นไทม์เมอร์หรือเคาน์เตอร์เพิ่มขึ้น

ไมโครคอนโทรลเลอร์เบอร์ที่นับเป็นเบอร์พื้นฐานสำหรับตระกูล MCS-51 นี้ได้แก่เบอร์ 8051,8031,8751 **โดยเบอร์ 8051** จัดเป็นสมาชิกตัวแรกของตระกูล วึ่งมีหน่วยความจำสำหรับเก็บโปรแกรมภายในชิปเป็น ROM ขนาด 4 กิโลไบต์ และหน่วยความจำสำหรับเก็บข้อมูลทั่วไปภายใน MCS-51 (RAM) เองจำนวน 128 ไบต์ มีพอร์ตขนาน 8 บิต 4 พอร์ต มีรีจิเตอร์สำหรับไทม์เมอร์หรือเคาน์เตอร์ขนาด 16 บิตรวม 2 ตัว รับสัญญาณอินเตอร์รัปต์จากภายนอก ได้ 2 ชนิด สามารถรับและส่งข้อมูลแบบอนุกรมผ่านทางพอร์ตสื่อสารข้อมูลแบบอนุกรม มีวงจรออสซิลเลเตอร์เพื่อสร้างสัญญาณนาฬิกาควบคุมการทำงานในตัวเอง

ส่วนเบอร์ 8751 จะมีคุณสมบัติเหมือนเบอร์ 8051 ทุกอย่าง ต่างกันชนิดหน่วยความจำสำหรับเก็บโปรแกรมภายในชิปของเบอร์ 8751 จะเป็น EPROM แทนที่จะเป็น ROM **ส่วนเบอร์ 8031** จะเหมือนกับเบอร์ 8051 ต่างกันเพียงในเบอร์ 8031 ไม่มีหน่วยความจำสำหรับเก็บโปรแกรมภายในชิป

ไมโครคอนโทรลเลอร์ตระกูล MCS-51 ทุกเบอร์ใช้แรงดันไฟเพียง 5 โวลต์ในการทำงาน ส่วนกระแสไฟฟ้าที่ใช้จะแตกต่างกันไปตามชนิดของเทคโนโลยีที่ใช้ในการผลิต เบอร์ไมโครคอนโทรลเลอร์ตระกูลนี้ที่มีอักษร C อยู่ตรงกลางเบอร์เช่น 80C31,80C51 จะเป็นเบอร์ของชิปที่ผลิตโดยอาศัย เทคโนโลยี CHMOS ซึ่งใช้พลังงานในการทำงานน้อยกว่าและสามารถควบคุมการใช้พลังงานของชิปได้จากโปรแกรมเพื่อการประหยัดพลังงานในระบบ

MCS-51 เป็นตระกูลไมโครคอนโทรลเลอร์ที่ถูกพัฒนาขึ้นมาจากตระกูล MCS-48 ดังนั้นจึงมีความสามารถเหนือกว่าหลายอย่าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 ตำแหน่งขาของ MCS-51

ไมโครคอนโทรลเลอร์ตระกูล MCS-51 ทุกเบอร์จะมีตำแหน่งขาพื้นฐานที่เหมือนกัน ดังรูป 3.1

หน้าที่การใช้งานแต่ละขาของชิปไมโครคอนโทรลเลอร์ในตระกูล MCS-51 มีดังนี้

- ขา Vss (ขา 20) สำหรับต่อลงกราวด์
- ขา Vcc (ขา 40) สำหรับต่อแหล่งจ่ายแรงดังกระแสตรงขนาด 5 โวลต์ (DC 5 volt)
- ขาพอร์ต 0 (ขา 32-39) มี 8 ขา ใช้เป็นขาสำหรับพอร์ต 0 ขนาด 8 บิต(**P0.0- P0.7**) แบบ Open Drain Bidirectional พอร์ตนี้สามารถใช้งานเป็นอินพุตเอาต์พุตพอร์ตทั่วไปได้ โดยหากใช้งานเป็นอินพุตพอร์ต ต้องโหลดค่า 1 ไปยังแต่ละบิตของพอร์ตนี้เพื่อบังคับให้ขาอยู่ในสถานะถูกปล่อย (มีสถานะ **high impedance**) นอกจากใช้งานเป็นอินพุตเอาท์พุตแล้ว พอร์ต 0 ยังใช้ในการติดต่อหน่วยความจำสำหรับเก็โปรแกรมและข้อมูลภายนอกชิปด้วย โดยส่งค่าแอดเดรสไบต์ต่ำ (A0-A7) และมัลติเพลกซ์กับการส่งข้อมูล (D0-D7) จากหน่วยความจำภายนอก ในระหว่างการเขียนหรือการอ่านข้อมูลโดยมีวงจรพูลอัพภายใน
- ขาพอร์ต 1 (ขา 1-8) มี 8 ขา ใช้เป็นขาสำหรับพอร์ต 1 (P1.0 - P1.7) สามารถใช้งานเป็นอินพุตหรือเอาต์พุตพอร์ตทั่วไปได้ หากต้องการใช้งานเป็นอินพุตพอร์ต ต้องโหลดค่า 1 ไปยังแต่ละบิตของพอร์ตนี้ เพื่อให้มีสถานะ high impedance โดยมีวงจรพูลอัพภายในขา P1.0 , P1.1 ในเบอร์ 8052 จะใช้งานในหน้าที่อย่างอื่นนอกเหนือจากใช้เป็นอินพุตเอาต์พุตทั่วไปด้วย
- ขาพอร์ต 2 (ขา 21-28) มี 8 ขา ใช้เป็นขาสำหรับพอร์ต 2 (P2.0-2.7) ขนาด 8 บิตแบบ Open Drain Bidirectional พอร์ตนี้สามารถใช้งานเป็นอินพุตเอาต์พุตพอร์ตทั่วไปได้โดยหากใช้งานเป็นอินพุตพอร์ต ต้องโหลดค่า 1 ไปยังแต่ละบิตของพอร์ตนี้ เพื่อบังคับให้ขาอยู่ในสถานะ high impedance นอกจากจะใช้งานเป็นอินพุตเอาต์พุตทั่วไปแล้ว พอร์ต 2 ยังใช้ในการติดต่อหน่วยความจำสำหรับเก็บโปรแกรมและข้อมูลภายนอกด้วย โดยใช้สำหรับส่งค่าแอดเดรสบัสไบต์สูง (A8-A15) และมีวงจรพูลอัพภายใน
- ขาพอร์ต 3 (ขา 10-17) มี 8 ขา ใช้เป็นขาสำหรับพอร์ต 3 (**P3.0-P3.7**) สามารถใช้งานเป็นอินพุตพอร์ตทั้วไปได้ หากต้องการใช้งานเป็นพอร์ตอิพุตพอร์ต ต้องโหลดค่า 1 ไปยังแต่ละบิตของพอร์ตนี้ เพื่อให้มีสถานะ **high impedance** โดยใช้วงจรพูลอัพภายใน นอกจากนี้ยังใช้งานในหน้าที่พิเศษต่างๆอีกหลายอย่างดังนี้

ขา P3.0 ใช้รับข้อมูลจากภายนอกแบบอนุกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขา P3.1 ใช้ส่งข้อมูลออกไปภายนอกแบบอนุกรม

ขา P3.2 ใช้เป็นอินพุตเพื่อรับสัญญาณอินเตอร์รัปต์ชนิดที่ 0

ขา P3.3 ใช้เป็นอินพุตเพื่อรับสัญญาณอินเตอร์รัปต์ชนิดที่ 1

ขา P3.4 สัญญาณอินพุตให้เคาน์เตอร์ของไทม์เมอร์ 0

ขา P3.5 สัญญาณอินพุตให้เคาน์เตอร์ของไทม์เมอร์ 1

ขา P3.6 ใช้เป็นสัญญาณควบคุมการเขียนข้อมูลการเขียนข้อมูลไปยังหน่วยความจำสำหรับเก็บข้อมูลภายนอกชิป

ขา P3.7 ใช้เป็นสัญญาณควบคุมการอ่านข้อมูลการเขียนข้อมูลไปยังหน่วยความจำสำหรับเก็บข้อมูลภายนอกชิป

การใช้งานพอร์ต ในหน้าที่พิเศษดังนี้กล่าวนี้จะต้องโหลดค่า 1 ไปยังแต่ละบิตที่ต้องการใช้ก่อนทุกครั้ง

- ขา RST (ขา 9) ใช้สำหรับการรีเซตวงจรทุกอย่างภายในชิป เพื่อเริ่มต้นการทำงานใหม่การรีเซต ใช้เมื่อเริ่มจ่ายพลังหรือเมื่อโปรแกรมทำงานผิดพลาด เมื่อต้องการรีเซตชิป MCS-51 ขานี้ต้องการรีเซตชิป MCS-51 ขานี้ต้องมีสถานะ 1 เป็นเวลาอย่างน้อย 2 แมชชีนไซเกิลระหว่างที่ออสซิลเลเตอร์ยังทำงานอยู่ โดยต้องต่อตัวต้านทานค่า 8.2 กิโลโอห์มเพื่อทำหน้าที่พูลดาวน์ (รักษาค่าแรงดันไฟฟ้าให้มีสถานะเป็นกราวด์) และเพื่อให้ตัวชิปรีเซตเองเมื่อเริ่มจ่ายพลังงานให้ต่อตัวเก็บประจุขนาด 10 ไมโครฟารัดคร่อมระหว่างขา RST กับ Vcc

- ขา ALE/PROG (ขา 30) เป็นขาสำหรับใช้ส่งสัญญาณออกไปภายนอก เพื่อควบคุมการแลตช์ค่าแอดเดรสไบต์ต่ำ จากพอร์ต 0 ในระหว่างการติดต่อหน่วยความจำสำหรับเก็บโปรแกรมหรือข้อมูลภายนอก ปกติเมื่อไม่มีการติดต่อหน่วยความจำภายนอกขานี้จะส่งสัญญาณพัลส์ออกมาด้วยความถี่ 1/8 ของความถี่ออสซิลเลเตอร์ที่ใช้ตลอดเวลา ดังนั้นเราสามารถใช้ความถี่ที่ได้จากขานี้ไปใช้งานอย่างอื่นได้ แต่ความถี่ที่ขานี้จะลดลงครึ่งหนึ่งในระหว่างติดต่อกับหน่วยความจำสำหรับเก็บข้อมูลที่อยู่ภายนอกชิป นอกจากนี้ขา ALE ยังใช้สำหรับควบคุมการเขียนโปรแกรมลงใน EPROM สำหรับ MCS-51 เบอร์ที่มีหน่วยความจำสำหรับเก็บโปรแกรมลงภายในเป็น EPROM

- ขา PSEN (ขา 29) ใช้ส่งสัญญาณสโตรบเพื่ออ่านคำสั่งจากโปรแกรมที่เก็บไว้ในหน่วยความจำภายนอกชิป เมื่อชิปทำงานด้วยโปรแกรมจากภายนอกขานี้จะส่งสัญญาณสโตรบสองครั้งในแต่ละแมชชีนไซเกิล แต่ในช่วงการเขียนหรืออ่านข้อมูลกับหน่วยความจำภายนอกหรือเมื่อใช้โปรแกรมจากหน่วยความจำสำหรับเก็บโปรแกรมภายในชิปจะไม่มีสัญญาณออกมาจากนี้

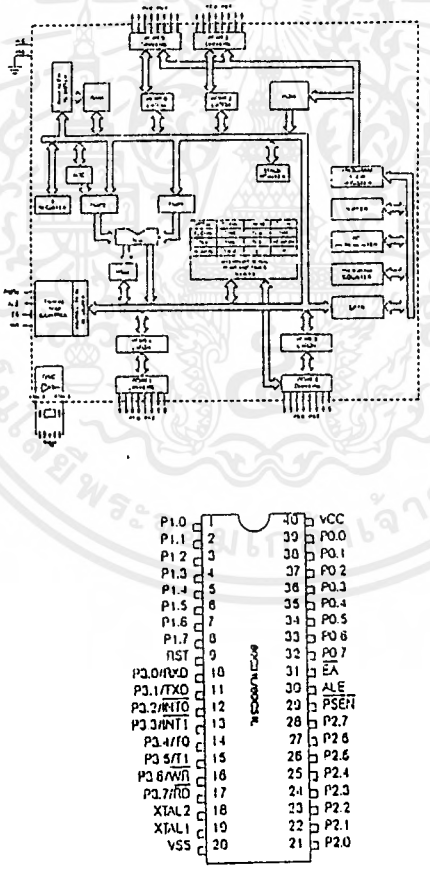
- ขา EA/Vpp (ขา 31) เป็นขาสำหรับใช้เลือกให้ MCS-51 ทำงานจากโปรแกรมที่อยู่ภายในหรือภายนอกชิป โดยหากขานี้มีสถานะเป็น 0 หมายถึงให้ใช้โปรแกรมจากหน่วยความจำที่เก็บ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมภายนอก หากขานี้มีสถานะเป็น 1 หมายถึงบังคับให้ MCS-51 ใช้โปรแกรมจากหน่วยความจำสำหรับเก็บโปรแกรมภายในชิป และสำหรับ MCS-51 ที่มีหน่วยความจำสำหรับเก็บโปรแกรมภายในชิปหรือจากโปรแกรมที่เก็บไว้ในหน่วยความจำภายนอกชิปด้วยการต่อขา EA กับไฟเลี้ยงหรือกราวด์ตามลำดับ ส่วนใน MCS-51 ที่ไม่มีหน่วยความจำสำหรับเก็บโปรแกรมภายในชิป ให้การต่อขานี้ลงกราวด์เสมอ

- ขา XTAL 1 (ขา 19) ใช้การต่อคริสตอลภายนอก โดยเป็นอินพุตเข้าสู่วงจรออสซิลเลเตอร์
- ขา XTAL 2 (ขา 18) ใช้การต่อคริสตอลภายนอก โดยเป็นอินพุตออกจากวงจรออสซิลเลเตอร์

3.3 สถาปัตยกรรมของ 8051



รูปที่ 3.1แสดงสถาปัตยกรรมภายในของ 8051 และการกำหนดขา DIP ของ 8051

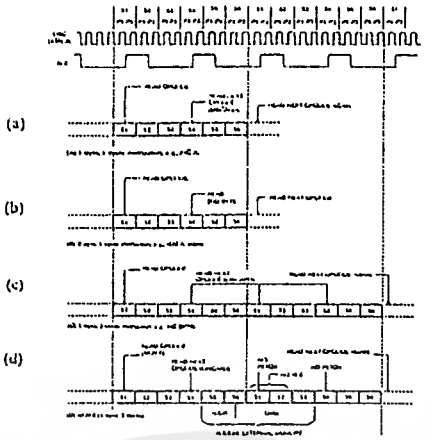
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4 การทำงานของ 8051

จากรูปที่ 3.1 เมื่อเริ่มป้อนไฟเลี้ยงให้กับ 8051 ซึ่งมีวงจร Power on reset ต่ออยู่จะมีการรีเซ็ทเกิดขึ้นการทำงานภายใน 8051 จะเริ่มจากบล็อก Program Counter ซึ่งเป็นวงจรนับ (Counter Circuit) ชนิดหนึ่งส่งค่าตำแหน่งหน่วยความจำสำหรับโปรแกรมไปบนบัส (Bus) หมายเลข 1 บัสนี้มีขนาด 16 บิท ค่าตำแหน่งหน่วยความจำนี้จะถูกส่งไปเก็บไว้ที่ Program ADDR Register ที่เป็นวงจร Latch ข้อมูลซึ่งเป็นค่าตำแหน่งหน่วยความจำ จะปรากฏที่บนบัส 16 บิทหมายเลข 2 ถ้าเป็นค่าตำแหน่งหน่วยความจำแรกหลังจากรีเซทค่าตำแหน่งหน่วยความจำจะเป็น 0000H หน่วยความจำสำหรับโปรแกรมจะเลือกได้ว่าเป็น ROM ภายในหรือภายนอก 8051 โดยการป้อนสภาวะลอจิกเข้าไปที่ 8051 ทางขา $\overline{EA}$ ซึ่งต่ออยู่กับส่วน Timing and Control ทำหน้าที่เป็นวงจรถอดรหัส (Decoder) แล้วสร้างสัญญาณควบคุมต่อไปถ้าป้อนสัญญาณลอจิก 1 เข้าไปที่ขา $\overline{EA}$ จะเป็นการเลือก ROM ภายใน 8051 โดยที่วงจร Timing and control จะสร้างสัญญาณไปยัง ROM ภายในให้ส่งข้อมูลที่เป็นคำสั่งจากตำแหน่งที่ถูกชี้ด้วยค่าตำแหน่งที่ส่งมาทางบัสหมายเลข 2 ข้อมูลจาก ROM จะถูกส่งลงไปยังบัสหมายเลข 3 ที่เรียกว่า Internal Data Bus แล้วนำไปเก็บไว้ที่ Instruction Register เพื่อส่งต่อไปให้กับวงจร Timng and Control ทำการถอดรหัสสัญญาณควบคุมการทำงานส่วนอื่น ๆ ต่อไปแล้วแต่ว่าเป็นคำสั่งให้ทำอะไร ในกรณีที่เลือก ROM ภายนอก 8051 โดยป้อนสัญญาณลอจิก 0 เข้าไปที่ขา $\overline{EA}$ จะทำให้วงจร Timing and Control ส่งสัญญาณไปยังพอร์ท 0 และ พอร์ท 2 เพื่อส่งตำแหน่งหน่วยความจำบนบัสหมายเลข 2 ออกไปชี้หน่วยความจำภายนอก จากนั้นจะอ่านข้อมูลที่เป็นคำสั่งกลับเข้ามาทางพอร์ท 0 ไปยัง Internal Data Bus แล้วไปเก็บที่ Instruction Register เพื่อทำงานต่อไปเหมือนกับตอนอ่านคำสั่งจาก ROM ภายใน การทำงานในช่วงส่งค่าตำแหน่งหน่วยความจำไปยังหน่วยความจำแล้วอ่านข้อมูลที่เป็นคำสั่งกลับเข้ามาเก็บไว้ใน Instruction Register เรียกว่าเป็นช่วงของการ Fetch ช่วงต่อไปจะเป็นช่วงของการทำงานตามคำสั่งเรียกว่า Execute Cycle

การทำงานที่กล่าวมาแล้วข้างต้นจะขึ้นกับสัญญาณควบคุมที่สร้างมาจากวงจร Timing and control และสัญญาณที่สร้างขึ้นนี้จะอ้างอิงกับสัญญาณนาฬิกาที่สร้างมาจาก Oscillator ทำให้การทำงานต่าง ๆ เป็นไปตามลำดับที่ผู้ผลิตได้ออกแบบไว้ ดังรูปที่ 3.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.2 ลำดับสถานะการทำงานใน MCS 51

คำสั่งแต่ละคำสั่งของ 8051 จะใช้เวลาทำงาน 1 , 2 หรือ 3 ไซเคิลของเครื่อง (Machine Cycle)แล้วแต่ว่าเป็นคำสั่งประเภทใด 1 ไซเคิลของเครื่องจะใช้เวลา 12 ไซเคิลของสัญญาณนาฬิกา ดังนั้นแต่ลรคำสั่งของ 8051 จะใช้เวลาการทำงาน 12 , 24 หรือ 36 ไซเคิลของสัญญาณนาฬิกานั่นเอง แต่ละไซเคิลของสัญญาณนาฬิกาจะถูกแบ่งเป็น 6 State คือ S1, S2, S3, S4, S5, S6 แต่ละ State จะประกอบด้วย 2 ไซเคิลของสัญญาณนาฬิกา ในไซเคิลแรกจะเรียกว่าเฟส 1 (P1) และไซเคิลที่ 2 เรียกว่าเฟส 2 (P2) ในแต่ละเฟสจะนับตั้งแต่ขอบขาลงของสัญญาณนาฬิกาถึงขอบขาลงของสัญญาณนาฬิกาที่อยู่ถัดไปดังในรูป เมื่อ 8051 ทำงานเสร็จ 1 ไซเคิลของเครื่องก็จะเริ่มทำงาน State 1 Phase 1 (S1P1) ของไซเคิลต่อไป ใน 1 ไซเคิลของเครื่องจักร Timing and Control จะสร้างสัญญาณ ALE ออกมา 2 ไซเคิลเพื่อ Fetch คำสั่งเข้าไป 2 ครั้งเสมอ ที่บริเวณขอบขาขึ้นของสัญญาณ ALE คำสั่งใดจะมีกี่ไบท์หรือใช้เวลาทำงานกี่ไซเคิลจะดูได้จากตารางชุดคำสั่ง 8051

คำสั่งประเภท 1 ไบท์ 1 ไซเคิลของเครื่องได้แก่คำสั่ง INC A จะมีการอ่านคำสั่งจากหน่วยความจำสำหรับโปรแกรม 2 ครั้ง ที่เวลาประมาณขอบขาขึ้นของสัญญาณ ALE เมื่อคำสั่งแรกถูกอ่านเข้าไปที่เวลาที่เวลาขอบขาขึ้นของ ALE แรก แล้วนำไปเก็บที่ Instrucrion Registor เพื่อให้วงจร Timing and Control ถอดรหัสแล้วเข้าอยู่การ Execute ขณะเดียวกันก็จะเริ่มต้นการ Fetch คำสั่งที่อยู่ในหน่วยความจำตำแหน่งถัดไปเข้ามาและคำสั่งที่ 2 จะถูกอ่านเข้ามาที่เวลาของขาขึ้นของสัญญาณ ALE ถัดไป วงจร Timing and Control เมื่อถอดรหัสคำสั่งแรกก็

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะทราบว่าการทำงานของคำสั่งนี้ให้สิ้นสุดจะใช้คำสั่ง เพียง 1 ไบท์ ดังนั้นคำสั่งที่ถูกอ่านมาไบท์ที่ 2 จะไม่ถูกนำมาทำงาน เพียงแต่อ่านเข้ามาแล้วทิ้งไป

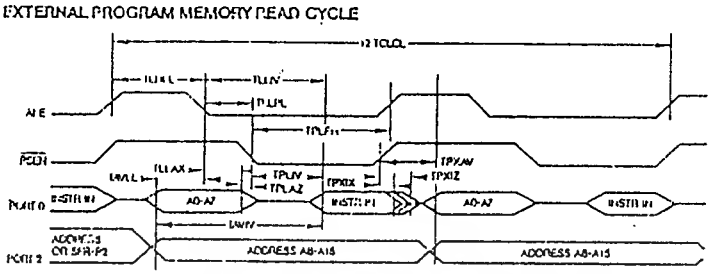
คำสั่งประเภท 2 ไบท์ และใช้เวลา 1 ไซเคิ้ลของเครื่องได้แก่คำสั่ง ADD A,#data ในหนึ่งไซเคิลของเครื่องนี้จะมีการอ่านคำสั่งเข้า 2 ไบท์ เหมือนกับคำสั่งประเภท 1 ไบท์ 1 ไซเคิลของเครื่อง แตกต่างกันที่ไบท์ที่ 2 จะถูกนำมาใช้ด้วยไม่ได้ถูกทิ้งไปดังในรูปที่ 10 b ตัวอย่างของคำสั่ง ADD A,#33H จะเขียนเป็นภาษาเครื่องได้ 2 ไบท์ คือ 2433 เมื่ออ่านคำสั่งไบท์แรกคือ 24 เข้าไปไว้ที่ Instruction Register แล้ว Timing and control จะถอดรหัสพบว่าเป็นคำสั่งบวกเลข ก็จะส่งสัญญาณไปยัง Accumulator ให้เอาข้อมูลไปไว้ที่ TMP1 คำสั่งที่ 2 ถูกอ่านเข้ามาที่ Instruction registor แล้ว Timing and control จะสั่งให้เอาข้อมูลที่ 2 ส่งไปยัง Internal Data bus ไปเก็บยัง TMP1 จากนั้นวงจร ALU จะนำเอาข้อมูล TMP1 และ TMP2 มารวมกัน ผลลัพธ์ที่ได้จะส่งออกจาก ALU ไปยัง Internal Data Bus แล้วไปเก็บที่ Accumulator

คำสั่งประเภท 1 , 2 หรือ 3 ไบท์ ที่ใช้เวลาทำงาน 2 ไซเคิลของเครื่องเช่นคำสั่ง INC DPTR จะมีการอ่านคำสั่งเข้าไป 4 ครั้งทุก ๆ ขอบขาขึ้นของสัญญาณ ALE ที่มี 2 ครั้งต่อ 1 ไซเคิลของเครื่อง ถ้าเป็นคำสั่งประเภท 1 , 2 หรือ 3 ไบท์ วงจร Timing and Control จะเอาคำสั่ง 1, 2, หรือ 3 ไบท์แรกเท่านั้นไปทำงานส่วนคำสั่งที่เหลือทิ้งไปดังรูปที่ 10 C คำสั่ง 1 ไบท์ที่ใช้เวลาทำงาน 2 ไซเคิลของเครื่องที่กล่าวมาแล้วจะไม่รวมถึงคำสั่ง MOVX ซึ่งใช้ในการอ่านหรือเขียนข้อมูลกับหน่วยความจำ Data Memory ภายนอก การทำงานของคำสั่งนี้จะมีการ Fetch คำสั่งเข้าไป 2 ไบท์ในไซเคิลของเครื่องแรก ในไซเคิลของเครื่องที่ 2 จะไม่มีการ Fetch คำสั่งเข้าไปแต่จะเป็นช่วงเวลาของการอ่านหรือเขียนข้อมูลกับ Data memory ภายนอก สัญญาณ ALE ซึ่งปกติจะเปลี่ยนเป็น 1 ที่ S1P2 ก็จะไม่เปลี่ยนเป็น 1 ในไซเคิลของเครื่องที่ 2 โดยจะเป็น 0 อยู่จนกว่า จะถึงเวลา S4P2 ของไซเคิลของเครื่องที่ 2 สัญญาณ ALE จะเปลี่ยนเป็น 1 เพื่อทำการอ่านหรือเขียนข้อมูลกับ Data Memory ภายนอก

3.5 ไดอะแกรมเวลาของการติดต่อกับหน่วยความจำ

การอ่านข้อมูลจากหน่วยความจำสำหรับโปรแกรมภายนอก 8051 นั้น สำดับสัญญาณตามเวลา (Timing Diagram) ของสัญญาณที่ทำการอ่านคำสั่งมีดังรูปที่ 3.3

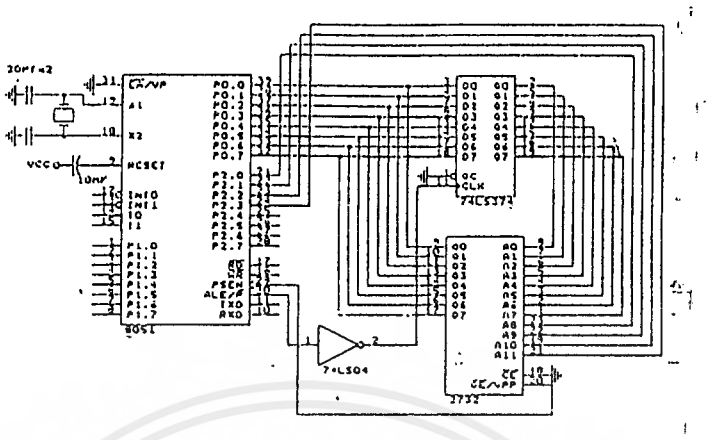
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.3 Timing diagram ของการอ่านโปรแกรมจากหน่วยความจำภายนอก

การอ่านคำสั่ง (Fetch) จาก Program area ภายนอกจะเริ่มจาก 8051 ส่งสัญญาณลอจิก 1 ออกมาทางขา ALE -ขณะนี้สัญญาณที่ขา PSEN จะเป็น 1 จากนั้น Port 0 จะส่งค่าตำแหน่งหน่วยความจำ 8 บิทล่างและพอร์ท 2 จะส่งตำแหน่งหน่วยความจำ 8 บิทบนออกมาแล้ว สัญญาณ ALE จะกลับเป็น 0 อุปกรณ์ภายนอกจะสามารถใช้ขอบขาลงของสัญญาณ ALE เพื่อ Latch ตำแหน่งหน่วยความจำที่พอร์ท 0 ไว้ จากนั้นพอร์ท 0 ก็จะยกเลิกการส่งค่าตำแหน่งหน่วยความจำเข้าสู่สภาวะ High Impedance และสัญญาณ PSEN เป็น 0 เพื่อเตรียมรับคำสั่งออกจากหน่วยความจำภายนอกเข้าไปยัง 8051 เพื่อทำงานต่อไป เมื่อคำสั่งถูกอ่านเข้าไปเก็บใน Instruction register แล้วสัญญาณ PSEN จะกลับเป็น 1 พร้อมกับสัญญาณ ALE ก็จะกลับเป็น High เพื่อการอ่านคำสั่งต่อไปทำงาน ข้อมูลในพอร์ท 2 จะคงที่ตลอดเวลา ตั้งแต่สัญญาณ ALE ก็จะกลับเป็นสัญญาณ ALE เปลี่ยนเป็น 0 และกลับเป็น 1 อีกครั้งหนึ่งจากนั้นจะเริ่มลำดับการ Fetch ข้อมูลไบท์ที่ 2 จากหน่วยความจำสำหรับโปรแกรม ซึ่งจะมีการเปลี่ยนแปลงของสัญญาณตามเวลาเหมือนกับการ Fetch ข้อมูลไบท์ที่ 2 จากหน่วยความจำสำหรับโปรแกรม ซึ่งจะมีการเปลี่ยนแปลงของสัญญาณตามเวลาเหมือนกับ Fetch ไบท์แรกนั้นเอง จาก Timing Diagram ดังกล่าวจะออกแบบวงจรที่มี Program Memory อยู่ภายนอก 8051 ได้ดังตัวอย่างในรูปที่ 3.4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.4 วงจรที่มี Program memory อยู่ภายนอก 8051

74LS374 ในรูปที่ 3.4 จะทำหน้าที่ Latch ตำแหน่งในหน่วยความจำ 8 บิทล่างที่เวลาขอบขาลงของสัญญาณ ALE ซึ่งสัญญาณ ALE จะถูกกลับให้เป็นตรงข้ามโดย Inverter 74LS04 ก่อนที่จะป้อนให้กับขา CK ของ 74LS374 และที่ขอบขาขึ้นของสัญญาณที่ออกจาก 74LS04 จะ Latch ตำแหน่งหน่วยความจำ ข้อมูลที่ออกจาก 74LS374 จะเป็นค่า 8 บิทล่างของตำแหน่งหน่วยความจำที่ต้องการติดต่อ ในวงจรได้ต่อค่าตำแหน่งหน่วยความจำ 8 บิทล่างของตำแหน่งหน่วยความจำที่ต้องการติดต่อ ในวงจรได้ต่อค่าตำแหน่งหน่วยความจำ 8 บิทเข้ากับ A0 ถึง A7 ของ EPROM และข้อมูลจากพอร์ท 2 บิท P2.0-P2.0 จะต่อเข้ากับ A8-A11 ของ EPROM โดยตรงเพราะค่าตำแหน่งหน่วยความจำ 8 บิทบนที่ออกมาจากพอร์ท 2 จะคงที่ตลอดเวลา ขา PSEN ของ 8051 จะถูกต่อเข้ากับขา OE ของ EPROM 2716 ดังนั้นเมื่อสัญญาณ PSEN มีสภาวะลอจิกเป็น 0 ก็จะส่งคำสั่งที่เก็บใน EPROM ณ ตำแหน่งที่ชี้โดยข้อมูลที่ขา A0-A11 ออกมายังพอร์ท 0 และถูก 8051 เก็บไปทำงานต่อไป

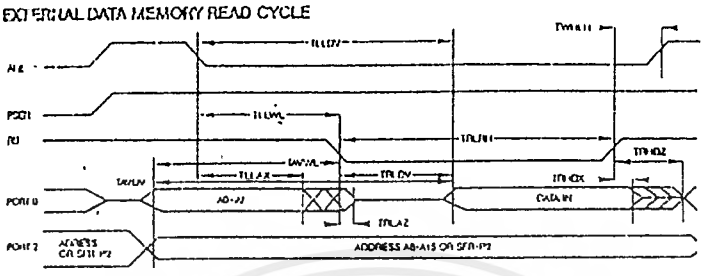
การอ่าน-เขียนข้อมูลกับหน่วยความจำสำหรับข้อมูลภายนอก 8051

การอ่าน-เขียนข้อมูลกับ Data Memory ภายใน 8051 นั้นจะมีสัญญาณสร้างมาจากส่วน Timing and Control โดยที่ผู้ใช้ไม่จำเป็นที่จะต้องทำความเข้าใจ แต่การอ่าน-เขียนข้อมูลกับ Data memory ภายนอก อันเนื่องมาจากคำสั่ง MOVX นั้น เมื่อคำสั่งดังกล่าวถูกอ่านเข้ามายัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Instruction Register แล้ว Timing and Control จะทำการถอดรหัสแล้วสร้างสัญญาณควบคุมดังนี้

การอ่านข้อมูลจาก External Data Memory จะมีไดอะแกรมสัญญาณตามเวลาดังรูปที่ 3.5



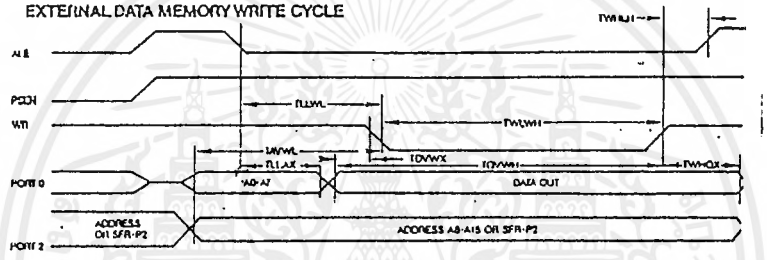
รูปที่ 3.5 Timing diagram -ของการอ่านข้อมูลจากหน่วยความจำสำหรับข้อมูล ภายนอก 8051

การทำงานจะเริ่มจากการส่งคาตำแหน่งหน่วยความจำภายนอก 8 บิทล่างออกทางพอร์ท 0 และ 8 บิทบนออกทางพอร์ท 2 เมื่อส่งค่าตำแหน่งแล้ว สัญญาณ ALE ซึ่งเดิมมีลอจิกเป็น 1 จะกลับมาเป็น 0 เพื่อให้อุปกรณ์ภายนอกสามรถ Latch ตำแหน่งหน่วยความจำไว้เหมือนกับในการอ่านข้อมูลจากหน่วยความจำสำหรับโปรแกรมภายนอก 8051 เพื่อส่งไปยังหน่วยความจำ แม้ว่าข้อมูลบนพอร์ท 0 จะเปลี่ยนแปลงไปก็ยังมีค่าตำแหน่งหน่วยความจำส่งไปยังหน่วยความจำในระหว่างการติดต่อกับ Data Memry นี้สัญญาณ PSEN จะเป็น 1 ตลอดเพราะสัญญาณ PSEN จะ Active (เป็น 0) ก็ต่อเมื่อเป็นการติดต่อกับหน่วยความจำสำหรับโปรแกรมภายนอก 8051 เท่านั้น 8051 จะส่งาญญาณลอจิก 0 ออกมาทางขา RD (P3.7) เพื่อบอกกับหน่วยความจำว่าต้องการอ่านข้อมูลเข้าไปเมื่อ 8051 ส่งสัญญาณ RD เป็นลอจิก 0 ยะทำให้พอร์ท 0 เข้าสู่สถานะ High Impedance พร้อมที่จะให้หน่วยความจำภายนอกส่งข้อมูลมาบนพอร์ท 0 ข้อมูลพอร์ท 0 ซึ่งส่งมาจากหน่วยความคำภายนอกจะถูกอ่านเข้าไปเก็บที่เวลาขอบขาขึ้นของสัญญาณ RD จากนั้นสัญญาณ ALE ก็จะกลับเป็น 1 เพื่อเริ่มการทำงานในคำสั่งต่อไปในระหว่างการอ่านข้อมูลจากหน่วยความจำสำหรับข้อมูลภายนอกนี้ พอร์ท 2 จะส่งค่าตำแหน่งหน่วยความจำ 8 บิทบนออกมาตลอกเวลา

การเขียนข้อมูลไปยังหน่วยความจำสำหรับข้อมูลภายนอก 8051 จะมีไดอะแกรมสัญญาณตามรูปที่ 3.6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

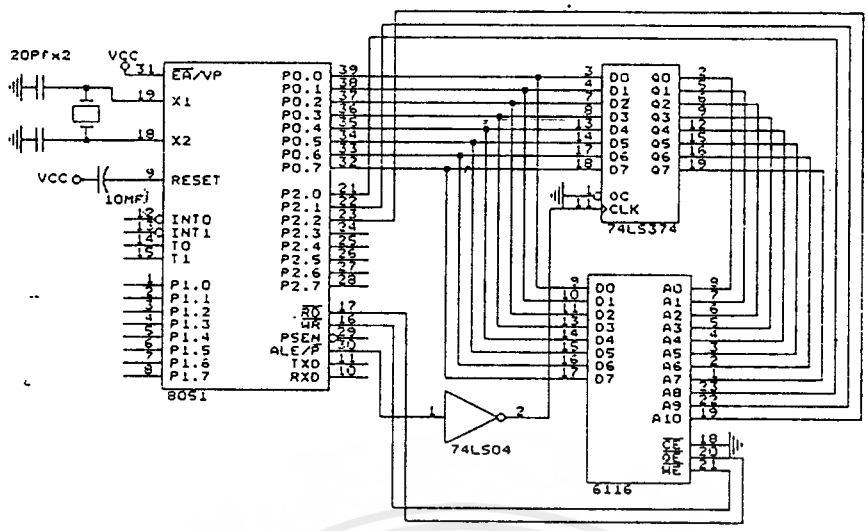
เมื่อ 8051 ส่งค่าตำแหน่งหน่วยความจำ 8 บิทล่างไปทางพอ์ท 0 และ 8 บิทบนลงไปทางพอร์ท 2 แล้ว สัญญาณ ALE จะกลับเป็น 0 อุปกรณ์ภายนอกจะสามรถใช้สัญญาณนี้ในการ Latch ค่าตำแหน่งหน่วยความจำบนพอร์ท 0 เหมือนกับในการอ่านข้อมูลจากหน่วยความจำสำหรับข้อมูลภายนอก เมื่อสัญญาณ ALE เป็น 0 แล้ว 8051 จะส่งข้อมูลที่ต้องการเขียนไปพอร์ท 0 แล้วจะให้สัญญาณ WR เปลี่ยนสภาวะลอจิกเป็น 0 ขณะนี้หน่วยความจำภายนอกจะต้องเขียนข้อมูลไปเก็บยังตำแหน่งที่กำหนด จากนั้นสัญญาณ WR จะกลับเป็น 1 เพื่อเป็นการบอกสิ้นสุดการเขียนข้อมูลแล้วสัญญาณ ALE ก็จะกลับเป็น 1 เพื่อ Fetch คำสั่งต่อไปมาทำงาน



รูปที่ 3.6 Timing diagram ของการเขียนข้อมูลไปยังหน่วยความจำสำหรับข้อมูลภายนอก 8051

หน่วยความจำสำหรับข้อมูลภายนอกที่สามารถอ่านและเขียนข้อมูลได้ จะสามารถเขียนเป็นวงจรได้ดังรูปที่ 3.7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.7 วงจรที่มีหน่วยความจำสำหรับข้อมูลที่อยู่ภายนอก 8051

74LS374 ในรูปจะใช้สำหรับ Latch ค่าตำแหน่งหน่วยความจำ 8 บิทล่างไว้ แม้ว่าข้อมูลบนพอร์ท 2 จะเปลี่ยนไป สัญญาณ RD และ WD จะอ่านหรือเขียนข้อมูลจากหน่วยความจำภายนอก 6116 เป็นหน่วยความจำแบบ RAM ที่สามารถจะอ่านและเขียนข้อมูลได้

3.6 TIMER Register THO,THI,TLI

ตำแหน่งหน่วยความจำภายในเท่ากับ 8CH, 8AH, 8DH, 8BH

ใน 8051 จะมีวงจร Timer อยู่ 2 ชุด คือ Timer 0 และ Timer 1 (8052 จะมี Timer 2 อีก 1 ชุด) ใน Timer แต่ละชุดจะมีรีจิสเตอร์ขนาด 8 บิทอยู่ 2 ตัว เพื่อเก็บค่าการนับของ Timer ได้สูงสุดถึง 16 บิทใน Timer 0 รี-จิสเตอร์นี้คือ TH0, TL0 และ ใน Timer 1 คือ รีจิสเตอร์ TH1, TL1 Tlx (x หมายถึง 0 หรือ 1) จะเก็บค่าของการนับ 8 บิทล่างและ Thx จะเก็บค่าของการนับ 8 บิทบน ผู้ใช้จะสามารถทำงานของวงจร Timer ในโหมด Timer หรือโหมด Counter ได้โดยการกำหนดในรีจิสเตอร์ชื่อ TMOD (Timer/Counter Mode Control Register) การทำงานเป็น Timer นั้นจะให้รีจิสเตอร์ใน Timer 0 หรือ 1 ทำการนับจำนวนไซเคิล (Cycle) ของสัญญาณนาฬิกาที่ผ่านวงจรหาร 12 ดังรูปที่ 3.9 เมื่อการนับครบถึงค่าสูงสุดที่รีจิสเตอร์ Tlx และ Thx จะเก็บได้ คือค่า FFFFH แล้วยังนับต่อไปค่าที่ได้จากการนับจะเป็น 0000H ทำให้เกิดการ Set บิทบางบิทในรีจิสเตอร์ TCON เพื่อบอกสภาวะ Timer Overflow นี้ในการให้วงจร Timer ทำงานเป็น Counter ก็คือการใช้รีจิสเตอร์

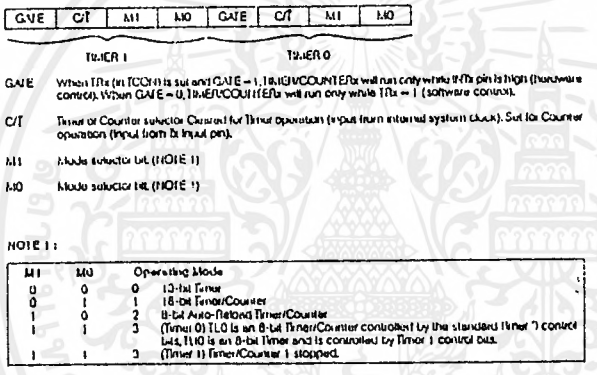
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Thx และ Tlx ทำการนับจำนวนไซเคิลของสัญญาณที่เข้ามาทางขา T0 หรือ T1 ขอ 8051 สัญญาณที่เข้ามาทางขา To หรือ T1 อาจจะมาจากอุปกรณ์ตรวจจับ (Sensor) ก็ได้แต่ละสภาวะของสัญญาณนี้จะต้องมีระดับโวลเตจของสภาวะลอจิก 0 หรือ 1 เป็นแบบ TTL คือลอจิก 0 จะต้องมีโวลเตจไม่เกิน 0.6 โวลท์ และลอจิก 1 จะต้องโวลเตทมากกว่า 2.4 โวลท์

3.6.1TMOD Timer / Counter mode register

ตำแหน่งหน่วยความจำภายในเท่ากับ 89H

TMOD เป็นรีจิสเตอร์ขนาด 8 บิท ที่มีหน้าที่ควบคุมการทำงานของ Timer 0 และ Timer 1 แต่ละบิทในรีจิสเตอร์นี้มีความหมายเฉพาะดังรูปที่ 3.8



รูปที่ 3.8 Tmod Timer / Counter Mode Register

ในรูป 3.8 M0 เป็นชื่อของบิท 0 และ GATE ทางซ้ายสุดเป็นชื่อของบิท 7 รีจิสเตอร์นี้แบ่งข้อมูลออกเป็น 2 ชุด ชุดละ 4 บิท คือ บิท 0-3 ใช้สำหรับควบคุมการทำงานของ Timer 0 และบิท 4-7 ใช้ควบคุมการทำงานของ Timer 1 หน้าที่ในการควบคุม Timer ของแต่ละบิทที่มีชื่อเดียวกันจะเหมือนกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

GATE เป็นบิทที่ใช้ควบคุมให้ Timer ทำงานหรือไม่ ถ้าบิทนี้ของ Timer x ถูกตั้งเป็น 1 จะทำให้ Timer ทำงานก็ต่อเมื่อที่ขา INTx มีสภาวะลอจิกเป็น 1 และบิท Trx ในรีจิสเตอร์ TCON เป็น 1 ด้วย

C / T บิทนี้ใช้สำหรับเลือกการทำงานของ Timer ว่าจะใช้เป็น Timer หรือ Counter ถ้าบิทนี้เป็น 1 ก็หมายความว่าเลือกการทำงานเป็น Counter ซึ่งจะนับจำนวนไซเคิลของสัญญาณที่เข้ามาทางขา Tx

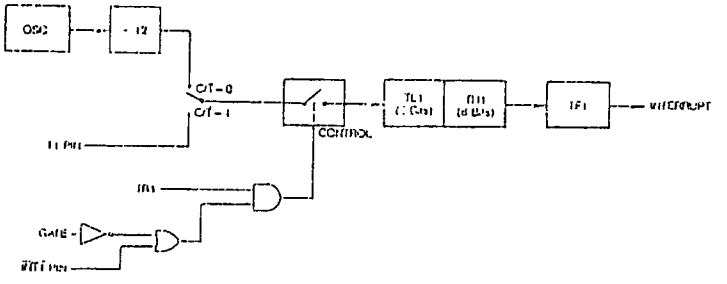
M1 , M0 เป็น 2 บิทที่ใช้ร่วมกันเพื่อเลือกโหมดการทำงานของ Timer การทำงานโหมด 0, 1 และ 2 ของ Timer 0 จะเหมือนกับ Timer 1 แต่ในโหมด 3 การทำงานของทั้งสองจะต่างกัน ค่าใน M1 และ M0 จะเลือกโหมดการทำงานดังนี้

M1	M0	การทำงาน
0	0	โหมด 0 รีจิสเตอร์ THx และ TLx ทำตัวเป็นตัวนับ 13 บิท ค่าจากการนับ 8 บิทบนมาจาก 8 บิทของ THx และอีก 5 บิทล่างมาจากค่า 5 บิทล่างของรีจิสเตอร์ TLx โดยที่ 3 บิทบน ของ TLx จะไม่ต้องสนใจเลย
0	1	โหมด 1 รีจิสเตอร์ THx และ TLx ทำตัวเป็นตัวนับ 16 บิทค่าจากการนับ 8 บิทบนอยู่ในรีจิสเตอร์ THx และค่าจากการนับ 8 บิทล่างอยู่ในรีจิสเตอร์ TLx
1	0	โหมด 2 ในการนับของรีจิสเตอร์ TLx ขนาด 8 บิทเมื่อนับถึงค่าสูงสุดคือ FFH เมื่อทำการนับต่อไปจะเกิดการ OverFlow แล้วก็จะ "Reload" เอาข้อมูลจาก THx เข้าไปยัง TLx เพื่อเป็นค่าเริ่มต้นในการนับครั้งต่อไป
1	1	โหมด 3 การทำงานของ Timer 0 และ Timer 1 จะต่างกันดังที่จะกล่าวต่อไป

การทำงานในแต่ละโหมดจะมีรายละเอียดดังนี้

โหมด 0

โหมด 0



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.9 Timer Mode 0 : 13 bit count

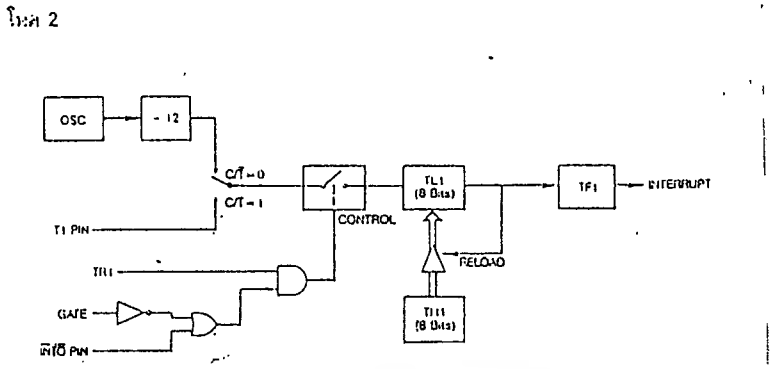
รูปที่ 3.9เป็นไดอะแกรมของวงจร Timer ภายใน 8051 ที่ทำงานในโหมด 0 ซึ่ง Timer 0 และ 1 ก็จะมีการทำงานเหมือนกันทุกประการ ในการอธิบายนี้จะใช้วงจรของ Timer 1 จากรูปจะเห็นสวิทช์ C/T ซึ่งถ้ากำหนดค่าในบิท C/T ของ TMOD เป็น 0 จะทำให้สวิตช์อยู่ในตำแหน่งบน เพื่อให้สัญญาณนาฬิกาความถี่ 12 เมกกะเฮิรตซ์ก็จะมีสัญญาณความถี่ 1 เมกกะเฮิรตซ์ออกจากวงจร หาร 12 ถ้าบิท C/T เป็น 1 จะทำให้สวิตช์ C/T อยู่ในตำแหน่งข้างล่าง เพื่อให้สัญญาณที่เข้ามาทาง T1 (หรือ T0 ถ้าเป็น Timer 0) ผ่านไปยังสวิตช์ Control สัญญาณที่เข้ามายังสวิตช์ Control จะส่งผ่านไปยังวงจรนับหรือไม่ก็ขึ้นอยู่กับสัญญาณควบคุมที่ออกมาจาก AND GATE ถ้าบิท TR1 (หรือ TR0 ถ้าเป็น Timer 0) ในรีจิสเตอร์ TCON เป็น 0 จะทำให้สภาวะของสัญญาณที่ออกจาก AND GATE เป็น 0 เสมอ และจะไม่มีสัญญาณใดออกจากสวิตช์ Control ไปยังวงจรนับเลย รีจิสเตอร์ TL1 และ TH1 จะไม่ทำงาน แต่ถ้าบิท GATE เป็น 0 หรือสัญญาณที่ขา INT1 (หรือ INT0 ถ้าเป็น Timer 0) และข้อมูลที่บิท GATE ของรีจิสเตอร์ TMOD ถ้าบิท GATE เป็น 0 หรือสัญญาณที่ขา INT1 มีสภาวะลอจิกเป็น 1 จะทำให้สัญญาณควบคุมสวิตช์ Control เป็น 1 ทำให้มีสัญญาณออกไปยังตัวนับรีจิสเตอร์ TL1 และ TH1 (หรือ TL0 และ TH0 ถ้าเป็น Timer 0) รีจิสเตอร์ TL1 จะทำการนับโดยมีการนับเพียง 5 บิทเท่านั้น (ทำหน้าที่เป็นวงจร Prescaler ขนาด 5 บิท) ซึ่งนับได้ตั้งแต่ 0 ถึง 31 เมื่อ TL1 นับสัญญาณที่ออกจากสวิทตช์ Control ครบ 32 ไซเคิลจะมีสัญญาณส่งไปยัง TH1 ไซเคิล บิท 5 ถึง 7 ของ TIx ที่ไม่ได้ใช้งานก็จะไม่ต้องสนใจ การทำงานของ Timer 0 และ 1 ในโมดนี้จะเหมือนกับการทำงานของ Timer ในไมโครคอนโทรลเลอร์เบอร์ 8048 ทุกประการ

โหมด 1

ในโหมดนี้จะมีการทำงานของวงจรภายในของ Timer 0 หรือ 1 เหมือนกับโหมด 0 ทุกประการแตกต่างกันที่ TIx จะถูกใช้งานทั้ง 8 บิท ทำให้ผลการนับใน TIx และ THx จะมีถึง 16 บิท

โหมด 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.10 Timer mode 2

ในรูปที่ 3.10 เป็นไดอะแกรมของวงจร Timer 1 ใน 8051 ที่ทำงานโมด 2 Timer 0 และ 1 มีการทำงานในโหมด 2 เหมือนกันโดยจะสามารถกำหนดให้ทำหน้าที่เป็น Timer หรือ Counter ได้โดยบิท C/T และควบคุมการนับได้โดยข้อมูลในบิท TR1 และ GATE ในรีจิสเตอร์ TMOD กับสัญญาณที่เขา INTx เมื่อเริ่มการทำงานข้อมูลในรีจิสเตอร์ TH1 จะถูกโหลด (Load) ไปยังรีจิสเตอร์ TH1 และ TL1 มีค่าเหมือนกัน เมื่อเกิดการนับจำนวนไซเคิลของสัญญาณที่ออกจากสวิตซ์ Control จะทำให้ค่าจากการนับในรีจิสเตอร์ TL1 เพิ่มขึ้นไปเรื่อย ๆ ทีละ 1 จนถึง OFFH ในการนับครั้งต่อไปจะทำให้บิท TF1 ในรีจิสเตอร์ TCON ไม่เป็น 1 และข้อมูลในรีจิสเตอร์ TH1 จะถูกโหลดไปยังรีจิสเตอร์ TL1 เพื่อเป็นค่าเริ่มต้นการนับต่อไป

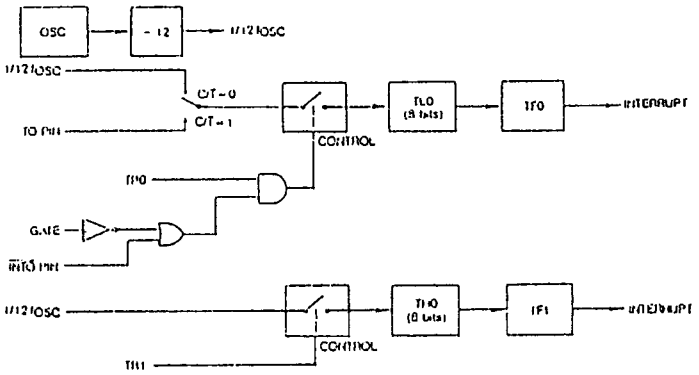
โหมด 3

การทำงานโหมด 3 ของ Timer 0 และ 1 จะต่างกัน

Timer 1 ในโหมด 3 จะไม่ทำงาน

Timer 0 ในโหมด 3 จะทำงานเป็นตัวนับที่เสมือนมีตัวนับ 8 บิทอยู่ 2 ตัว คือ TL0 และ TH0 ทำงานแยกกันดังรูปที่ 3.11

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.11 Timer 0 mode 3

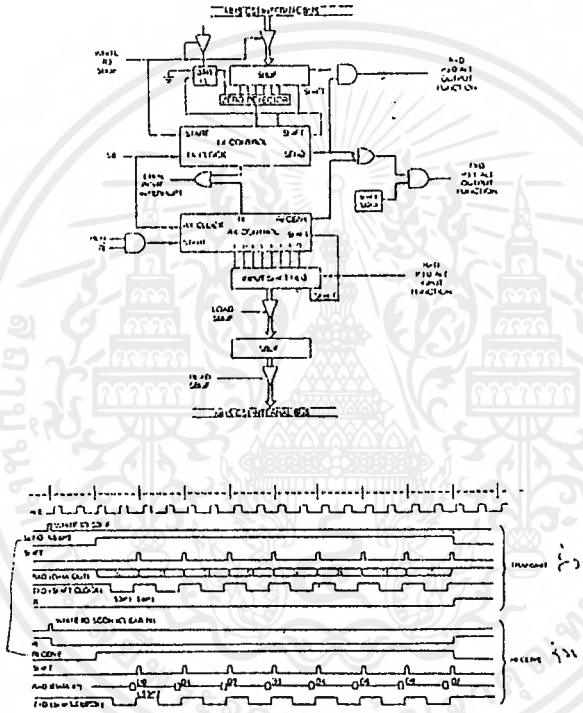
รีจิสเตอร์ TL0 จะเป็นตัวนับ 8 บิท ที่มีการนับสัญญาณจากออสซิลเลเตอร์หารด้วย 12 หรือนับสัญญาณที่เข้ามาทางขา T0 ขึ้นกับบิท C/T ในรีจิสเตอร์ TMOD และการนับจะควบคุมโดยบิท TR0 และ GATE ในรีจิสเตอร์ TMOD กับสภาวะลอจิกของสัญญาณที่ขา INT0 เหมือนกับในการทำงานโหมด 0, 1 และ 2 แต่ค่าจากการนับนี้สูงสุดจะมีเพียง 255 เท่านั้น เมื่อค่าการนับเปลี่ยนจาก 0FFH เป็น 00H คือเกิดการ overflow จะทำให้บิท TF0 ถูก SET เป็น 1 และอาจเกิดการขัดจังหวะ (Interrupt) การทำงานของโปรแกรมได้ถ้ามีการกำหนดค่าในรีจิสเตอร์ IE และ IP

ตัวนับอีกตัวคือรีจิสเตอร์ TH0 จะทำงานในโหมดของ Timer เท่านั้น คือจะนับจำนวนไซเคิลของสัญญาณที่ออกจากออสซิลเลเตอร์แล้วหารด้วย 12 การนับจะควบคุมได้ด้วยบิท TR1 ในรีจิสเตอร์ TMOD ถ้า
บิทนี้เป็น 1 ก็จะมีเข้าไปยัง TH0 แต่ถ้าบิทนี้เป็น 0 ก็จะไม่มีสัญญาณเข้าไปยัง TH0

3.7 การรับ-ส่งข้อมูลทางพอร์ทอนุกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการรับ-ส่งข้อมูลแบบอนุกรมผ่านทางพอร์ทอนุกรมนั้น จะต้องมีการกำหนดโหมดการทำงานในรีจิสเตอร์ SCON และในบางโหมดของการทำงานจะสามารถกำหนดอัตราการส่งข้อมูลได้โดยการโปรแกรมใน Timer ข้อมูลที่จะส่งออกหรือรับเข้าทางพอร์ทอนุกรมจะอยู่ที่รีจิสเตอร์ SBUF การทำงานของวงจรภายในแต่ละโหมดมีดังนี้



รูปที่ 3.12 Serial port mode 0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.12 เป็นไดอะแกรมที่ทำงานในการรับ-ส่งข้อมูลทางพอร์ทอนุกรม โหมด 0 และ ไดอะแกรมสัญญาณตามแวลาของสัญญาณที่จุดต่าง ๆ ในวงจร

การทำงานโหมด 0

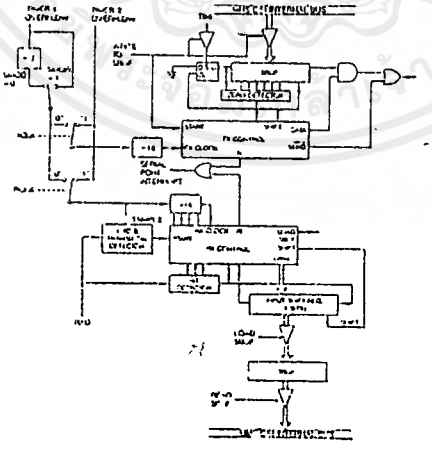
การส่งข้อมูล

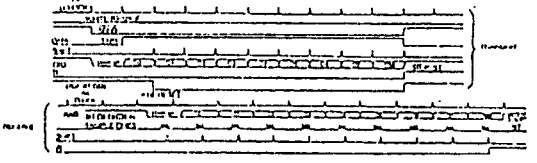
การส่งข้อมูลจะเริ่มจากการทำงานของคำสั่งให้เคลื่อนย้ายข้อมูลเข้ามายังรีจิสเตอร์ SBUF โดยจะมีสัญญาณ Write to SBUF ซึ่งเกิดขึ้นในช่วงเวลา S6P2 สัญญาณนี้จะทำให้ข้อมูลจากบัสภายใน 8051 (8051 Internal Bus) ทางด้านบนถูกนำไปเก็บที่รีจิสเตอร์ SBUP อีกทั้งยังทำการ Set ให้ D FLIF-FLOP ที่อยู่ทางซ้ายของ SBUF มีสภาวะลอจิกที่ขา Q เป็น 1 สัญญาณ Write to SBUF ซึ่งต่อเข้าไปยังขา Start ของ TX Control จะบอกให้ TX Control เริ่มส่งข้อมูลโดยจะหน่วงเวลาไว้ 1 ไซเคิลของเครื่องเมื่อผ่านเวลา 1 ไซเคิลของเครื่องไปแล้วขึ้นไซเคิลใหม่ สัญญาณ SEND จะเปลี่ยนสภาวะลอจิกจาก 0 เป็น 1 และเริ่มส่งข้อมูลบิท 0 จากนี้ในทุกเวลา S6P2 สัญญาณ SHIFT จะเปลี่ยนเป็น 1 ออกจากวงจร TX Control (ที่เวลาอื่นสัญญาณนี้จะเป็น 0) ทำให้ข้อมูลใน SBUF ถูกเลื่อนออกไปทีละบิทตรงขอบขาลงของสัญญาณ Shift ในทุก ๆ ไซเคิลของเครื่อง ขณะที่ข้อมูลถูกเลื่อนออกไปทางขวานี้ ข้อมูลจาก D-FF จะเลื่อนเข้ามาทางซ้าย ข้อมูลจาก D-FF บิทแรกที่เลื่อนเข้ามาจะเป็น 1 เพราะถูก Set ไว้ตอนแรก แต่บิทต่อมาจะเป็น 0 เพราะขา D ถูกต่อไว้ที่กราวด์ ขณะที่สัญญาณ SEND มีลอจิกเป็น 1 ทำให้เอาท์พุทของ OR Gate มีลอจิกเป็น 1 ดังนั้นสัญญาณจากวงจร Shift Clock จะถูกส่งออกไปทางขา P3.1 (TXD) โดยสัญญาณนี้จะเป็น 1 ในช่วงเวลา S6P1 ถึง S2P2 ของไซเคิลเครื่องถัดไปและ จะเป็น 0 ในช่วงเวลา S3P1 ถึง S5P2 ในไซเคิลของเครื่องเดียวกัน สัญญาณจากขา TXD ที่ส่งออกไปนี้ก็เพื่อให้อุปกรณ์ปลายทางสามารถรับข้อมูลได้ถูกต้องเพราะถูกส่งออกไปพร้อมกับข้อมูลใน SBUF ข้อมูลจำนวน 8 บิทจะถูกเลื่อนออกไปทางขา RXD จนครบทั้ง 8 บิท เมื่อข้อมูลบิทสุดท้ายถูกส่งออกไปจะทำให้ 1 ที่เกิดขึ้นจาก D-FF ในตอนเริ่มต้นการส่งข้อมูลถูกเลื่อนมาอยู่ทางขวาสุดของรีจิสเตอร์ SBUF และทางซ้ายทั้งหมดจะเป็น 0 ทำให้วงจร Zero Detector ซึ่งตรวจสอบค่า 0 นี้ส่งสัญญาณไปบอกวงจร Tx Control ว่าสิ้นสุดการส่งข้อมูลออก เมื่อสิ้นสุดการส่งข้อมูลสัญญาณ SEND จะเปลี่ยนสภาวะลอจิก 1 เป็น 0 ที่ขอบขาลงของสัญญาณ SHIFT และการส่งข้อมูล
บิทสุดท้ายออกไปจะทำให้บิท TI ในรีจิสเตอร์ SCON เปลี่ยนจาก 0 เป็น 1 บอกการสิ้นสุดของการส่งข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การรับข้อมูล

การรับข้อมูลทางพอร์ทอนุกรมในโมด 0 จะเริ่มต้นรับข้อมูลเข้ามาทางขา RXD ก็ต่อเมื่อมีการทำงานของคำสั่ง Set ค่าบิท-REN-เป็น 1 และ Clear บิท-RI เป็น 0 ในรีจิสเตอร์ SCON ซึ่งการทำงานของคำสั่งนี้จะทำให้เริ่มรับข้อมูลที่เวลา S6P2 เสร็จแล้วจะหน่วงเวลาไปจนถึง S6P2 ในไซเคิลของเครื่องถัดไปก็จะทำให้สภายะลอจิกของสัญญาณ Receive เปลี่ยนจาก 1 เป็น 0 เพื่อเริ่มส่งสัญญาณ Clock จากวงจร Shift Clock ถูกส่งออกไปทางขา TXD โดยสัญญาณ Clock ที่ขานี้จะมีสภาวะลอจิกเป็น 1 ตั้งแต่ S6P1 จนถึง S2P2 ของไซเคิลเครื่องถัดไปและมีสภาวะลอจิกเป็น 0 ในช่วงเวลา S3P1 ถึง S5P2 แต่ก่อนที่สัญญาณ Receive จะเปลี่ยนเป็น 1 นั้น วงจร RX Control จะเขียนข้อมูล 11111110H เข้าไปยัง Input Shift Register จากนั้นในทุก ๆ เวลา S6P2 ซึ่งสัญญาณ Shift มีสภาวะลอจิกเป็น 1 (นอกนั้นที่เวลาอื่นจะมีลอจิกเป็น 0) จะเลื่อนข้อมูลใน Input Shift Register ไปทางซ้ายทีละ 1 บิท เมื่อครบ 7 ครั้ง ข้อมูล 0 ซึ่งอยู่ทางขวาสุดของรีจิสเตอร์ในตอนเริ่มต้นจะเลื่อนมาอยู่ตำแหน่งซ้ายสุดและบอกให้กับ RX control ว่ามีข้อมูลบิทสุดท้ายอีก 1 บิทที่จะเข้ามา เมื่อข้อมูลสุดท้ายเข้ามาที่เวลา S5P2 ของไซเคิลของเครื่องที่ 9 (ข้อมูลที่เข้ามายังพอร์ทอนุกรม จะถูกอ่านเข้าไปที่เวลา S5P2 แต่จะถูกเก็บเข้าไปที่ Input shift register ที่เวลา S6) นับตั้งแต่เริ่มรับข้อมูลจะทำให้ที่เวลา S1P1 ของไซเคิลของเครื่องที่ 10 นับตั้งแต่เริ่มมีสัญญาณ Wrrite to SCON (Clear RI) จะเกิดการส่งข้อมูลจาก Input Shift Register ไปเก็บยังรีจิสเตอร์ SBUF ต่อไป และในขณะเดียวกันก็จะทำให้บิท RI ซึ่งถูก Clear ตั้งแต่เริ่มต้นรับข้อมูลถูก Set ให้เป็น 1 และจะทำให้สัญญาณ Receive เป็น 0 ทำให้ไม่มีสัญญาณ Clock ออกไป





เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูป 3.13 Serial port mode 1

ในโหมดนี้จะมีการส่งข้อมูลชุดละ 10 บิท คือ Start bit 1 บิท ข้อมูล 8 บิทและ Stop 1 บิท อัตราของการส่งข้อมูลในโหมดนี้จะขึ้นกับอัตราการเกิด Overflow ใน Timer 1 ข้อมูลนี้จะถูกส่งออก 1 บิททุก 16 หรือ 32 ครั้งของการเกิด Overflow ขึ้นใน Timer 1 (การ Overflow คือ การที่ข้อมูลใน Timer Register มีค่าเพิ่มขึ้นจนถึงค่าสูงสุด เมื่อเพิ่มค่าไปอีกก็จะกลับเป็น 0) ใน 8052 จะสามารถกำหนดให้อัตราการส่งข้อมูลขึ้นกับอัตราการเกิด Overflow ของ Timer 2 ได้ดังในรูปที่ 3.13

การทำงานของโหมด 1

การส่งข้อมูล

จากรูปที่ 3.13 บิท SMOD จะเป็นตัวเลือกว่า สัญญาณ Timer 1 Overflow ที่ส่งไปยังวงจรหาร 16 จะถูกหาร 2 ก่อนหรือไม่ ถ้า SMOD เป็น 1 สัญญาณ Timer 1 Overflow จะไม่ถูกหาร แต่ถ้า SMOD เป็น 1 สัญญาณ Timer Overflow จะถูกหาร 2 ก่อนที่จะเข้าวงจรหาร 16 การส่งข้อมูลจะเริ่มจากการที่มีคำสั่งเขียนข้อมูลไปยังรีจิสเตอร์ SBUF จะมีสัญญาณ Write to SBUF เกิดขึ้นเพื่อรับข้อมูลจาก Internal Bus ด้านบนไปเก็บยังรีจิสเตอร์ SBUF และทำให้เอาท์พุทของ D FLIP FLOP ทางซ้ายของ SBUF มีค่าเป็น 1 และเป็นบิทที่ 9 ของการส่งข้อมูลสัญญาณ Write to SBUF ยังส่งไปยัง TX control ด้วย ขณะนี้ข้อมูลในวงจรหาร 16 มีค่าเป็นอะไรไม่ทราบจึงจะรอจนกว่าข้อมูลในวงจรหาร 16 นับเพิ่มขึ้นจนถึงค่าสูงสุดแล้ววกกลับเป็น 0 คือเกิดการวนกลับทำให้เริ่มการส่งข้อมูลที่เวลา S1P1 ของไซเคิลเครื่องถัดไป (การส่งข้อมูลออกจะสัมพันธ์กับการเกิด Overflow ในวงจรหาร 16) สัญญาณ SEND จาก TX Control เปลี่ยนสภาวะลอจิกเป็น 0 แล้วเริ่มส่งข้อมูลที่เป็น Start bit (0) ออกไป เมื่อสง Start Bit ออกไปแล้ววงจร Tx Control ก็จะทำให้สัญญาณ DATA เป็น 1 เพื่อเลื่อนข้อมูลใน SBUF ออกไปเริ่มจากบิท 0 จนถึงบิทที่ 7 การส่งข้อมูลนี้จะเกิดขึ้นเมื่อสัญญาณ Tx Clock เปลี่ยนสถานะจาก 0 เป็น 1 ดังในรูปที่ 3.13 ขณะที่ข้อมูลถูกเลื่อนออกไปนั้นจะมี 0 อยู่ทั้ง 8 บิทใน SBUF ทำให้ Zero Detector รู้ว่าเป็นข้อมูลบิท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สุดท้ายแล้วที่ส่งออกโดยจะมีสัญญาณมาบอกกับวงจร Tx Control ด้วยเมื่อ TX Control ส่งสัญญาณ Shift ออกไปเป็นการส่งข้อมูลบิทสุดท้าย (บิท 7) ออกไป ก็จะรออีก 1 TX Clock (Bit Clock) ก็จะทำให้ขา TXD ส่งข้อมูล Stop Bit (1) ออกมา สัญญาณ DATA ซึ่งมีสภาวะลอจิกเป็น 1 มาตั้งแต่เริ่มส่งข้อมูลบิท 0 ก็จะกลับเป็น 0 และบิท TI จะเป็น 1 เพื่อบอกการสิ้นสุดการส่งข้อมูลทั้งหมดจะสิ้นสุดลงเมื่อสัญญาณ TX Clock ไซเคิลที่ 10 นับตั้งแต่สัญญา SEND เปลี่ยนสภาวะลอจิกเป็น 0

การรับข้อมูล

การรับข้อมูลจะขึ้นกับอัตราการเกิด Overflow ใน Timer 1 แล้วหาร 2 หรือไม่ขึ้นอยู่กับค่าของบิท SMOD สัญญาณนี้จะไปเข้าวงจรหาร 16 และเป็นตัวกำหนดอัตราการรับข้อมูลการรับข้อมูล จะเริ่มจากวงจร 1-TO-0 Transition Detector พบว่าสัญญาณที่เขา RXD เปลี่ยนจาก 1 เป็น 0 ซึ่งหมายถึงข้อมูล Start bit เข้ามาการตรวจสอบนี้จะกระทำด้วยอัตราเดียวกับสัญญาณที่เข้าวงจรหาร 16 เมื่อพบการเปลี่ยนสถานะลอจิกที่ขา RXD ก็จะเริ่มการรับข้อมูล ขณะนี้จะรีเซ็ทวงจรหาร 16 ให้มีค่าเป็น 0 เพื่อสร้างสัญญาณ RX Clock ให้เข้าจังหวะ (Synchronous) กับข้อมูลที่เข้ามาโดยสัญญาณ RX Clock จะเป็น 1 เมื่อการนับของวงจรหาร 16 มีค่าเป็น 15 ขณะที่วงจรหาร 16 นับถึง 7, 8 และ 9 จะมีการตรวจสอบข้อมูลที่เข้ามาทางขา RXD เพื่อเป็นการตรวจว่าข้อมูลนั้นเป็นอะไร ถ้าอย่างน้อยข้อมูล 2 ใน 3 เป็นค่าใดก็จะถือว่าข้อมูลที่เข้ามาเป็นค่านั้น ถ้าในการตรวจสอบ Start Bit แล้วพบว่าผิดพลาด คือไม่เป็น 0 ก็จะรีเซ็ทการทำงานเพื่อไปตรวจสอบการเปลี่ยนสถานะจาก 1 เป็น 0 ของข้อมูลที่ขา RXD ใหม่ แต่ถ้าพบ Start bit ก็จะเก็บข้อมูลทั้งหมดที่เข้ามาโดยเลื่อนข้อมูลเข้าไปยัง Input Shift Registar ก็มีสัญญาณควบคุมการเลื่อนข้อมูล (Shift) ส่งมาจาก RX control ในตอนเริ่มต้นการรับข้อมูลจะมีการเขียนข้อมูล 1FFH ไปเก็บใน Input Shift Register ขณะที่ข้อมูลถูกเลื่อนเข้าไปทางขวาของ Input Shift Register ก็จะมี 1 ถูกเลื่อนออกไปทางซ้ายทุกครั้งที่มีข้อมูลเข้ามา เมื่อ Stxart bit ที่รับเข้ามาถูกเลื่อนไปถึงซ้ายสุด Input Shift Register ก็จะมีสัญญาณไปบอก RX Control Block หลังจากข้อมูลบิทสุดท้ายเข้ามาแล้วก็จะโหลด (Load) เอาข้อมูล 8 บิทไปเก็บในรีจิสเตอร์ SBUF พร้อมทั้ง Set ค่าในบิท RI และ RB8 ของรีจิสเตอร์ SCON แต่การโหลดข้อมูลไปเก็บนี้จะเกิดขึ้นได้ก็ต่อเมื่อ

1. RI = 0 และ
2. SM2 = 0 หรือถ้า SM2 = 1 จะต้องได้รับ stop bit เป็น 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถ้าไม่มีสภาวะใดสภาวะหนึ่งดังกล่าวแล้ว ข้อมูลที่รับเข้ามาก็จะถูกทิ้งไปคือไม่โหลดไปเก็บในรีจิสเตอร์ SBUF ถ้ามีสภาวะดังกล่าวถูกต้อง stop bit จะถูกนำไปเก็บในรีจิสเตอร์ SBUF และบิท RI จะเป็น 1

แต่ไม่ว่าทั้ง 2 กรณีจะเกิดหรือไม่ก็จะกลับไปสู่การตรวจสภาวะเปลี่ยนจาก 1 เป็น 0 ที่ขา RXD
เพื่อข้อมูลต่อไป

$$\text{Baud rate} = \frac{2^{SMOD}}{32} \times (\text{Timer 1 Overflow Rate})$$

ในขณะที่ใช้ Timer 1 เป็นตัวกำหนด Baud Rate นี้จะต้อง Disable ไม่ให้เกิดการขัดจังหวะเนื่องมาจากการ Overflow Timer 1 อาจใช้ในโมดของ Timer หรือ Counter ก็ได้ ซึ่งเมื่อการนับในรีจิสเตอร์ตัวนับมีค่าสูงสุดแล้วกลับมาเป็น 0 ก็จะเกิด Overflow เช่นเดียวกัน แต่โดยปกติแล้วจะใช้ Timer 1 นี้ ในโมดของ Timer ที่มีการทำงานแบบ Auto Reload โมด 2 เพื่อว่าเมื่อค่าในการนับโดยรีจิสเตอร์ TL1 ถึงค่าสูงสุดก็จะโหลดค่าในรีจิสเตอร์ TH1 มาไว้ใน TL1 สำหรับเป็นค่าเริ่มต้นการนับต่อไป ซึ่ง Baud rate จะมีค่า

$$\text{Baud rate} = \frac{2^{SMOD}}{32} \times \frac{\text{Oscillator frequency}}{12 \times [\ 256 - (TH1)\ ]}$$

โดยที่ SMOD เป็นบิทหนึ่งในรีจิสเตอร์ PCON

เช่นความถี่ของออสซิลเลเตอร์เท่ากับ 11.059 Mhz บิท SMOD = 0 รีจิสเตอร์ TH1 มีค่า E8H, Timer 1 ทำงานในโมด 2 จะได้ว่าอัตราการส่ง-รับข้อมูลแบบอนุกรม

$= (2^{0}/32)*[(11.059*10^{6})/(12*(256-232))]$

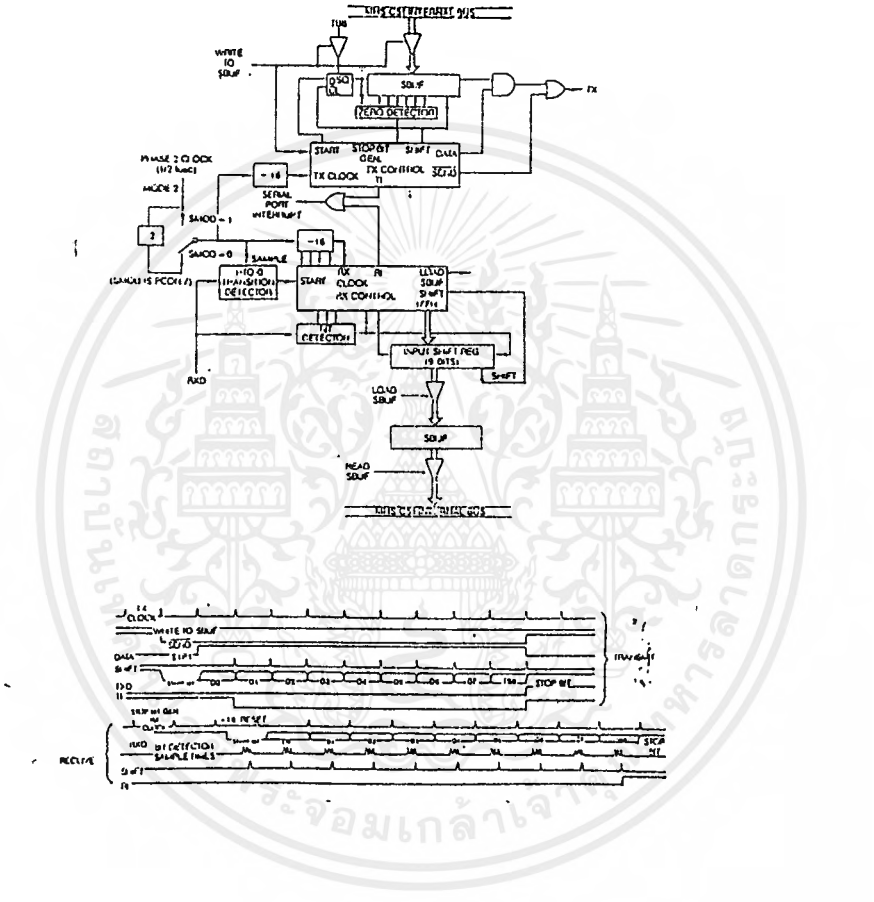
$=1200$ บิท/วินาที

การทำงานโหมด 2 และโหมด 3

โหมด 2 และโหมด 3 ของการรับ-ส่งข้อมูลทางพอร์ทอนุกรมจะเหมือนกันแตกต่างกันเฉพาะตรงที่อัตราการส่งข้อมูลเท่านั้น ในโหมด 2 จะเลือกอัตราการรับ-ส่งข้อมูลได้เป็น 1/32 หรือ 1/64 เท่าของความถี่สัญญาณออสซิลเลเตอร์ส่วนในโหมด 3 จะสามารถกำหนดอัตราการส่งข้อมูล

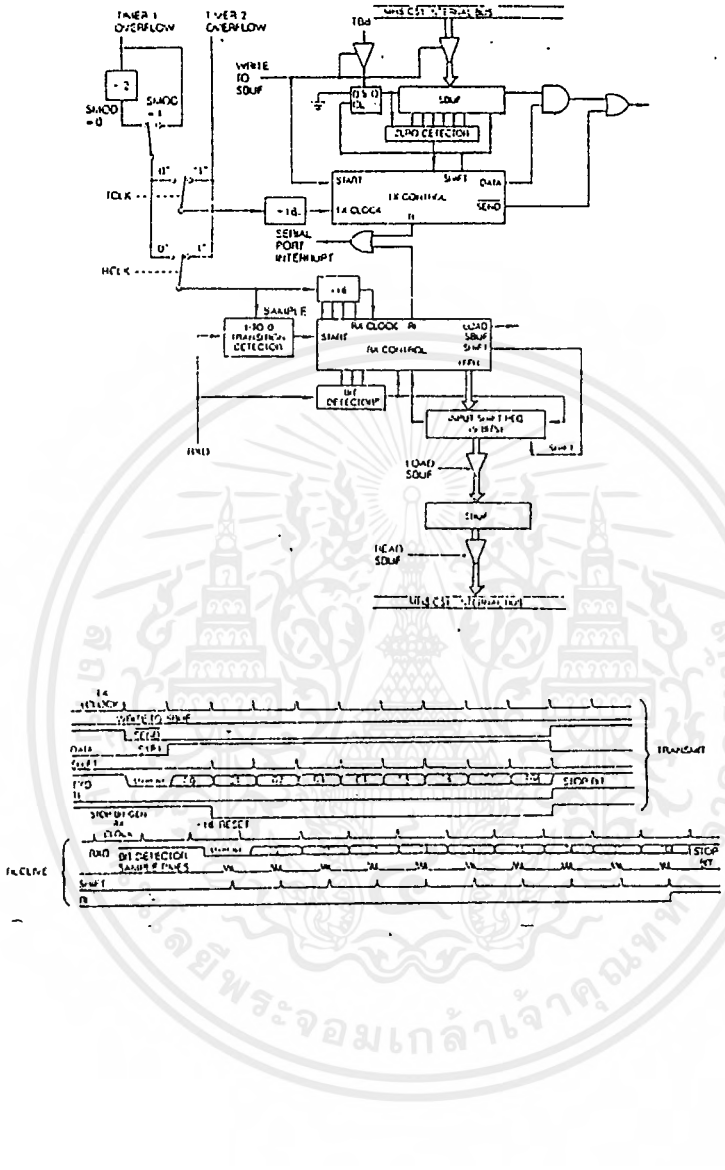
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ได้โดยอัตราการส่งข้อมูลจะขึ้นกับอัตราการเกิด Overflow ของ Timer เช่นเดียวกับการรับ-ส่งข้อมูลโหมด 1



รูปที่ 3.14 Serial port mode 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.15 Serial port mode 3

ในรูปที่ 3.14 จะเห็นว่าสัญญาณจากเฟส 2 ของสัญญาณนาฬิกาซึ่งมีความถี่เท่ากับ 1/2 ของสัญญาณจากออสซิลเลเตอร์จะถูกเลือกโดยสวิตช์ SMOD ว่าจะให้หาร 2 หรือ ไม่ สวิตช์ SMOD นี้จะสามารถกำหนดตำแหน่งให้อยู่ข้างบนหรือข้างล่างโดยการกำหนดค่าของบิท SMOD

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในรีจิสเตอร์ PCON ให้เป็น 1 หรือ 0 ตามลำดับ สัญญาณที่ออกจากวงจรหาร 16 จะมีความถี่เป็น 1/32 หรือ 1/64 เท่าของสัญญาณจากออสซิลเลเตอร์

ในรูป 3.15 สัญญาณ Timer 1 Overflow อันเกิดจากการนับถึงค่าสูงสุดใน Timer 1 แล้วค่าการนับจะกลับเป็น 0 ใหม่ สัญญาณนี้จะถูกเลือกโดยสวิตช์ SMOD ให้หาร 2 หรือไม่ก่อนที่จะส่งไปยังวงจรหาร 16 เพื่อสร้างเป็นสัญญาณควบคุมการเลื่อนข้อมูลเข้าหรือออกทางพอร์ทอนุกรมต่อไป การเลือกตำแหน่งของสวิตช์ SMOD ก็โดยการกำหนดค่าในบิท SMOD ของรีจิสเตอร์ PCON ถ้าบิทนี้เป็น 1 สัญญาณ Timer 1 Overflow จะถูกส่งไปยังวงจรหาร 16 โดยตรงแต่ถ้าบิทนี้เป็น 0 สัญญาณ Timer 1 Overflow จะถูกหาร 2 ก่อนแล้วจึงส่งเข้าไปยังวงจรหาร 16 การกำหนดอัตราการส่งข้อมูลจะทำให้เหมือนกับในการรับ-ส่งข้อมูลพอร์ทอนุกรมโมด 1 ที่กล่าวมาแล้วทุกประการ

การรับ-ส่งข้อมูลในโมด 2 และ 3 นั้น 1 ชุดของข้อมูลจะประกอบด้วย 1 Start Bit, 9 Data Bit 1 Stop Bit, ทั้งหมดนี้จะส่งออกหรือรับเข้าทางพอร์ทอนุกรมทีละบิทเรียงตามลำดับจากรูปที่ 3.14 และ 3.15 จะสามารถอธิบายการทำงานภายใน 8051 ในการส่งหรือรับข้อมูลได้ดังนี้

การส่งข้อมูล

การส่งข้อมูลจะเริ่มจากมีการเขียน (Write) ข้อมูลลงไปยังรีจิสเตอร์ SBUF ทำให้มีสัญญาณ Write to SBUF เกิดขึ้น แล้วข้อมูล 8 บิทจาก Internal Data Bus ของ 8051 ถูกเขียนลงไปในรีจิสเตอร์ SBUF และ ข้อมูลจากบิท TB8 ของรีจิสเตอร์ SCON จะเขียนลงไปยัง D FLIP FLOP เช่นกัน รวมเป็น 9 บิท ดังรูปที่เวลา S1P1 ของไซเคิลเครื่องแรกหลังจากสัญญาณ TX Clock นับครบ 1 รอบ จะทำให้สัญญาณ SEND ที่ออกจากวงจร TX Control เปลี่ยนสภาวะลอจิกเป็น 0 เพื่อเริ่มการส่งข้อมูลโดยจะเริ่มส่ง Star Bit (0) ออกไปทางขา TXD เมื่อสัญญาณ TX Clock เปลี่ยนเป็นๆ 1 ครั้งต่อไปก็จะเริ่มส่งข้อมูลบิท 0 ออกไปและสัญญาณ DATA จะเป็น 1 ที่เวลานี้ด้วย TX Control จะส่งสัญญาณ Shift ไปยังรีจิสเตอร์ SBUF ทุกครั้งที่สัญญาณ TX Clock เป็น 1 เพื่อเลื่อนข้อมูลออกไปทางขวา และจะเลื่อน 0 เข้ามาทางซ้ายขณะที่บิท TB8 อยู่ทางขวาสุด เตรียมส่งออกนั้นข้อมูลทางซ้ายทุกบิทจะเป็น 0 วงจร Zero Detector จะมีสัญญาณไปบอก TX Clock (Bit time) เมื่อส่งข้อมูลบิท TB8 ออกไปเสร็จสิ้นแล้วจะรอเวลา 1 Bit Time สัญญาณ SEND ก็จะกลับไปเป็น 1 และบิท T1 ในรีจิสเตอร์ SCON จะเป็น 1 เพื่อบอกสิ้นสุดการส่งข้อมูล และสัญญาณ DATA ก็จะกลับเป็น 0 จากนั้นสัญญาณที่ออกจาก TXD ก็คือ Stop Bit นั้นเอง

การรับข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การรับข้อมูลที่เข้ามาทางขา RXD ของ 8051 จะเริ่มต้นเมื่อวงจร 1-to-0 Transition Detector ทำตรวจสอบข้อมูลที่ขา RXD 16 เท่าของอัตราการส่งข้อมูลพบว่าสัญญาณที่ขา RXD เปลี่ยนจาก 1 เป็นทำให้วงจรหาร 16 ถูกรีเซ็ทไปด้วยและจะเขียนข้อมูลในรีจิสเตอร์ SBUF เป็น 1 FFH (มี 9 บิท) เมื่อ Counter ในวงจรหาร 16 นับถึง7,8 และ 9 จะมีการตรวจสอบข้อมูลที่เข้ามาทางขา RXD ถ้า 2 ใน 3 เหมือนกันก็ถือว่าข้อมูลที่เข้ามาคือค่านั้น ถ้าข้อมูลที่รับเข้ามาบิทแรกไม่เป็น 0 คือไม่ใช่ Start Bit (เพราะอาจตรวจพบการเปลี่ยนจากสัญญาณจากขา RXD แต่การตรวจสอบที่เวลา 7, 8 และ 9 ส่วนใหญ่เป็น 1) ก็จะเกิดการรีเซ็ทส่วนรับข้อมูลแล้วกลับไปเริ่มการตรวจสอบการเปลี่ยนสถานะจาก 1 เป็น 0 ใหม่แต่ถ้าถูกต้องก็จะรับข้อมูลเข้ามาทีละ 1 บิท ข้อมูลนี้จะถูกเลื่อนเข้าไปเก็บทางขวาของ Input Shift Register และ 1 จะถูกเลื่อนออกไปทางซ้ายโดยสัญญาณ Shift จนกระทั่ง

สัญญาณ Start Bit ซึ่งเป็น 0 ถูกเลื่อนเข้ามาถึงซ้ายสุดของ Inupt Shift Register จะบอกให้กับ RX Control รู้ว่าจะต้องมีข้อมูลบิทสุดท้ายเข้ามาอีก 1 บิท เมื่อข้อมูลบิทสุดท้ายเข้ามาจะเกิดการโหลดข้อมูลจาก Input Shift Register ไปยัง SBUF,RB8 และเซต R1 ให้เป็น 1 แต่สิ่งนี้จะเกิดขึ้นได้ต้องมีสภาวะดังนี้ด้วย

1. บิท R1 เป็น 0 และ
2. SM2 = 0 หรือถ้า SM2 = 1 ข้อมูลบิทที่ 9 จะต้องเป็น 1

ถ้าไม่มีสภาวะดังกล่าวทั้ง 2 ข้อมูลที่รับมาจะถูกทิ้งไปไม่ถูกนำไปเก็บที่ SBUF และ R1 ก็จะไม่ถูก Set แต่ถ้าเกิดขึ้น ทั้ง 2 กรณี ข้อมูลจะถูกนำไปเก็บที่รีจิสเตอร์ SBUF 8 บิท และบิทสุดท้ายจะนำไปเก็บที่บิท RB8 ของรีจิสเตอร์ SCON หลังจากนั้น 1 Bit Timer ไม่ว่าจะมีการเก็บข้อมูลหรือ ก็จะเข้าสู่การตรวจสอบสถานการเปลี่ยนสัญญาณ 1 เป็น 0 เพื่อเตรียมรับข้อมูลต่อไป

3.8 การขัดจังหวะ (Interrupt)

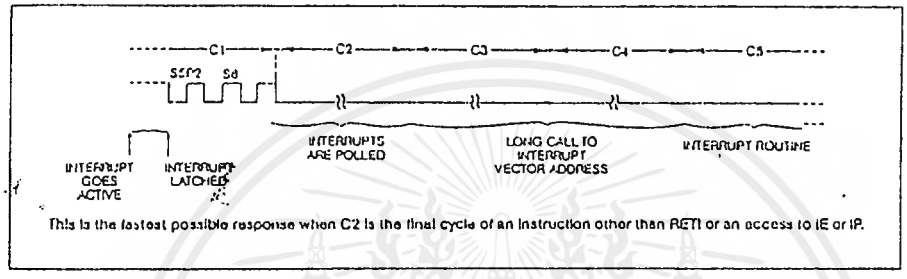
การขัดจังหวะคือสภาวะหนึ่งที่คอมพิวเตอร์กำลังทำงานอยู่แล้วถูกขัดจังหวะด้วยสัญญาณหรือคำสั่งพิเศษที่ทำให้คอมพิวเตอร์ต้องละจากงานที่กำลังอยู่ ไปทำงานในโปรแกรมตอบสนองการขัดจังหวะนั้นเมื่อเสร็จแล้วก็จะกลับมาทำงานเดิมต่อไปได้ ใน 8051 จะสามารถขัดจังหวะการทำงานได้ 6 แหล่ง คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. INTO, INT1 เป็น 2 ขาของ 8051 ที่จะรับสัญญาณจากภายนอก การขัดจังหวะจะเกิดขึ้นถ้าสัญญาณที่ขาดังกล่าวมีสภาวะลอจิกเป็น 0 หรือเปลี่ยนจาก 1 เป็น 0 โดยเลือกด้วยการกำหนดในบิท IT0 หรือ It1 ในรีจิสเตอร์ TCON

2. TF0, TF1 เป็นบิทหนึ่งที่จะบอกการทำงานของ Timer 0, Timer 1 เมื่อเกิด Overflow ขึ้นใน Timer จะทำให้บิทนี้เป็น 1 และเกิดการขัดจัวหวะการทำงานของ 8051 ได้

3. TI, RI เป็น 2 บิท ในรีจิสเตอร์ SCON ถ้าบิทนี้ถูก เซต ให้เป็น 1 โดยฮาร์ดแวร์ อันเนื่องจากเสร็จสิ้นการส่งหรือรับข้อมูลจะสามารถทำให้เกิดการขัดจังหวะได้



รูปที่ 3.16 ไดอะแกรมเวลาของการตอบสนองการขัดจังหวะ

8051 จะทำการอ่านสัญญาณจากทั้ง 6 แหล่งที่เวลา S5P2 ของทุก ๆ ไซเคิลของเครื่อง (Machine Cycle) เข้ามาเก็บและในช่วงของไซเคิลของเครื่องถัดไปก็จะตรวจสอบสถานะของสัญญาณทั้ง 6 ที่เก็บเข้ามา ถ้าสัญญาณนั้นมีการขัดจังหวะที่ถูกต้อง 8051 ก็จะละทิ้งการทำงานเดิมไว้ชั่วคราวแล้วสร้างคำสั่ง LCALL ขึ้นมาภายใน 8051 เพื่อไปทำงานในโปรแกรมตอบสนองการขัดจังหวะแต่ละสัญญาณนั้น เมื่อทำงานในโปรแกรมตอบสนองการขัดจังหวะเสร็จสิ้นก็จะสามารถกลับมาทำงานเดิมได้ โดยใช้คำสั่ง RETI เป็นคำสั่งสุดท้ายในโปรแกรมตอบสนองการขัดจังหวะ สัญญาณขัดจังหวะจากแต่ละแหล่งจะมีตำแหน่งหน่วยความจำที่จะเก็บโปรแกรมตอบสนองการขัดจังหวะไว้ต่างกันดังนี้

สัญญาณที่ขอขัดจังหวะ ตำแหน่งเริ่มต้นโปรแกรมตอบสนองการขัดจังหวะ

1	INT0	0003H
2	TF0	000BH
3	INT1	0013H

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4	TF1	001BH
5	TI,RI	0023H

ตำแหน่งเริ่มต้นโปรกแกรมนี้เป็นตำแหน่งใน Program area เช่น ถ้ามีสัญญาณ INT0 เข้ามาแล้ว 8051 ตรวจสอบว่ามีการขอขัดจังหวะถูกต้อง ก็จะละทิ้งการทำงานเดิม แล้วไปทำงานที่โปรแกรมตอบสนองการขัดจังหวะที่มีตำแหน่งเริ่มต้นอยู่ที่ตำแหน่ง 0003H เมื่อเสร็จสิ้นการทำงานของโปรแกรมตอบสนองการขัดจังหวะจะต้องมีคำสั่ง RETI อยู่เพื่อกลับมาสู่การทำงานเดิมได้ 8051 จะทำการตรวจสอบสัญญาณดังกล่าวว่ามีสัญญาณใดขอการขัดจังหวะมาบ้างโดยใช้วิธี Plolling คือการตรวจสอบเรียงตามลำดับจาก 1, 2, 3, 4 และ 5 ตามลำดับ ดังนั้น ถ้ามีการขอการขัดจังหวะเข้ามาพร้อม ๆ กัน 8051 ซึ่งตรวจสอบการขอขัดจังหวะของสัญญาณต้น ๆ ก่อน หรืออีกนัยหนึ่งคือ สัญญาณของการขัดจังหวะต้น ๆ จะมีลำดับความสำคัญสูงสุด (Highest Priority) และสัญญาณที่ 5 จะมีลำดับความสำคัญต่ำสุด (Lowest Priority) อย่างไรก็ตามสามารถที่จะจัดลำดับความสำคัญของสัญญาณขัดจังหวะนี้ใหม่เพื่อให้มีการตอบสนองการขัดจังหวะสัญญาณขอการขัดจังหวะลำดับหลังได้ โดยการโปรแกรมในรีจิสเตอร์ IP (Interrupt Priority Register) และจะสามารถกำหนดว่าจะให้ทำโปรแกรมตอบสอนงการขัดจังหวะ เมื่อมีสัญญาณขอขัดจังหวะเข้ามาหรือไม่ก็ได้ โดยการโปรแกรมในรีจิสเตอร์ IE (Interrupt Enalbe Register)

เมื่อ 8051 ทำการตรวจสอบสัญญาณขอการขัดจังหวะที่เก็บเข้ามาเมื่อเวลา S5P2 ฯ แล้วพบว่ามีการขอขัดจังหวะนั้น แม้ว่าจะมีการ Enable ในรีจิสเตอร์ IE ถูกต้อง แต่จะต้องมีเงื่อนไขดังนี้ด้วย

1. ไม่ได้กำลังทำงานในโปรตอบสนองการขัดจังหวะของสัญญาณขัดจังหวะที่มีลำดับความสำคัญสูงกว่าหรือเท่ากัน เช่น กำลังทำงานในโปรแกรมตอบสนองการขัดจังหวะของสัญญาณ INT0 อยู่แล้วมีการขอขัดจังหวะจากสัญญาณ INT1 อีก จะไม่เกิดการทิ้งงานเดิม คือไม่มีการไปทำงานที่โปรแกรมตอบสนองการขัดจังหวะของสัญญาณ INT1

2. เนื่องจากการสุ่มสัญญาณเข้าไปเพื่อตรวจสอบนั้นจะทำที่เวลา S5P2 ของในไซเคิลสุดท้ายของคำสั่ง และคำสั่งที่อยู่ถัดมาจะต้องใช้เวลาทำงาน 2 ไซเคิลของเครื่อง ดังนั้นการตรวจสอบจะกระทำในไซเคิลแรก แม้ว่าจะมีการขอการขัดจังหวะเข้ามาก็จะไม่ทำให้โปรแกรมตอบสนองการขัดจังหวะ จะต้องอ่านสัญญาณที่เวลา S5P2 อีกครั้งแล้วไปตรวจสอบที่ไซเคิลที่ 2 คำสั่ง ถ้ามีการขอขัดจังหวะถูกต้องจึงจะข้ามไปทำงานในโปรแกรมตอบสนองการขัดจังหวะ

3. คำสั่งที่กำลังทำงานอยู่ขณะที่ตรวจสอบสัญญาณขอขัดจังหวะ จะต้องไม่ใช่คำสั่ง RET หรือคำสั่งใด ๆ ก็ตามที่พยายามเขียนข้อมูลไปยังรีจิสเตอร์ IE หรือ IP

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สัญญาณขอขัดจังหวะที่ถูกอ่านเข้าไปที่เวลา S5P2 นี้ไม่ว่าได้รับการตอบสนองหรือไม่ ก็จะถูกทิ้งไป
แล้วอ่านเข้าไปใหม่ทุกเวลา S5P2

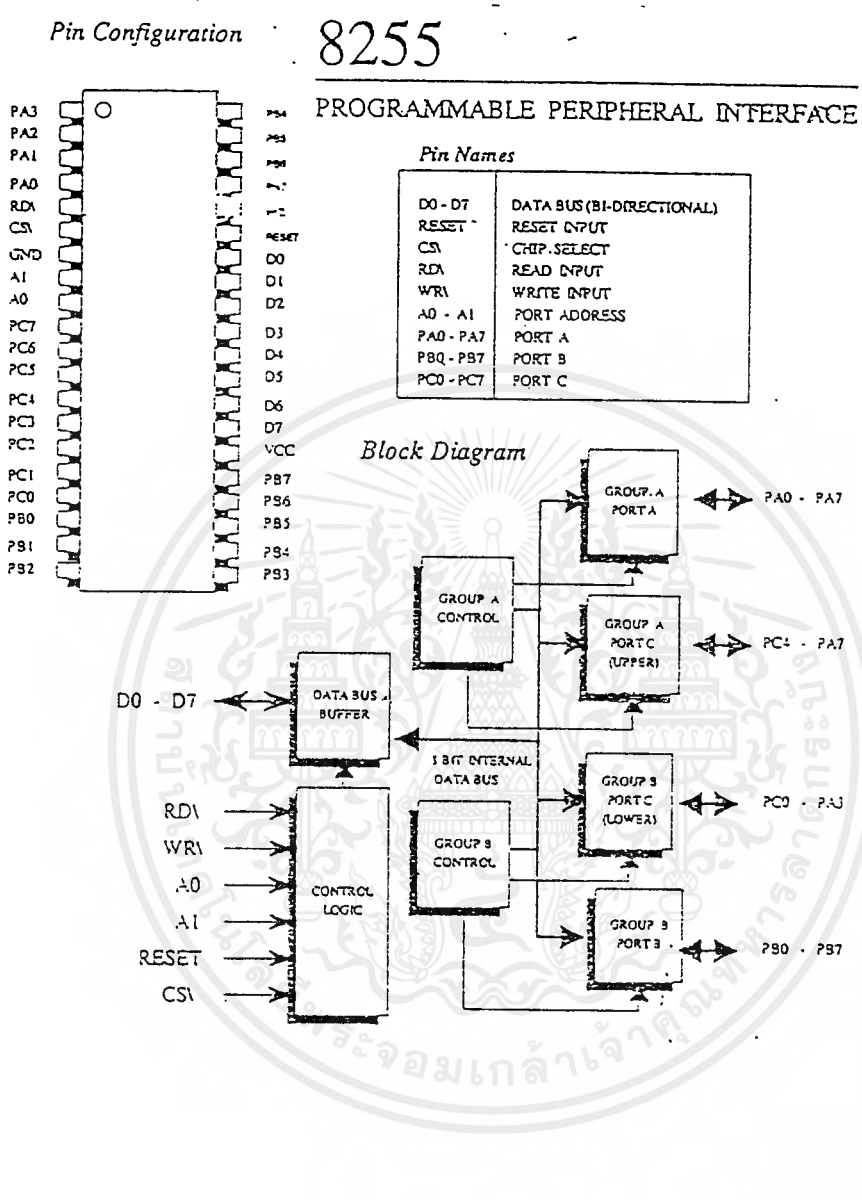
3.9 การใช้งาน 8255 กับ 8051

การขยายเพิ่มเติมพอร์ตอินพุท/เอาท์พุทพอร์ตของ 8051 นอกจากใช้ไอซีประเภทแลตช์และบัฟเฟอร์ประกอบกันเข้ากับบัสของระบบแล้ว ยังสามารถใช้ไอซีวงจรรวมความจุสูง (หรือ LSI) เบอร์ 8255 ซึ่งสามารถทำหน้าที่ได้ทั้งพอร์ตอินพุทหรืออินพุทตามโปรแกรมด้วยซอฟต์แวร์ ทำให้มีความอ่อนตัวในการนำไปใช้งานตามความประสงค์มากขึ้น ต่อไปนี้จะได้กล่าวแนะนำถึงลักษณะพื้นฐานของไอซีนี้ รวมถึงการโปรแกรมใช้งานและการเชื่อมต่อเข้ากับบัสของ 8051 ซึ่งจะทำให้สามารถนำไปประยุกต์ใช้ได้อย่างเหมาะสมต่อไป
ลักษณะพื้นฐานของ 8255

ไอซีเบอร์ 8255 ได้รับการออกแบบมาเพื่อเป็นพอร์ต สำหรับการรับ/ส่งข้อมูลแบบขนานระหว่างอุประกรณ์ภายนอกกับไมโครคอนโทรลเลอร์ ความอ่อนตัวในการนำไปใช้งานของ 8255 ได้แก่ การที่สามารถเปลี่ยนแปลงลักษณะการทำงานของพอร์ต ให้เป็นการเอาท์พุทหรืออินพุทได้สะดวก เพียงการส่งข้อมูลควบคุมจากไมโครคอนโทรลเลอร์ก่อนที่จะเริ่มต้นใช้งานเท่านั้น ความสามารถเช่นนี้เรียกว่า programmable คือ สามารถโปรแกรมการทำงานได้ ทำให้ได้รับความนิยมนำไปใช้งานกันอย่างแพร่หลาย

จากแผนภาพในรูปที่ 3.17 จะเห็นได้ว่า 8255 ประกอบด้วยบล็อกของหน่วยการทำงานหลายส่วนอยู่ภายในบล็อกทางขวามือจำนวน 4 บล็อก เป็นส่วนที่เชื่อมต่อกับอุปกรณ์ภายนอกโดยตรงผ่านทางเส้นสัญญาณที่ระบุชื่อว่า PA0-PA7,PB0-PB7 และ PC0-PC7 กลุ่มของสัญญาณเหล่านี้จำแนกเป็น 3 กลุ่ม คือ พอร์ต A , B และ C สำหรับบล็อกถัดเข้ามาบริเวณส่วนกลางที่มีชื่อว่า Group A Control และ Group B Control ทำหน้าที่กำหนดการทำงานของพอร์ททั้งสาม (ซึ่งจะได้อธิบายต่อไป) บล็อกทั้งสองนี้เชื่อมต่อกับบล็อกอื่นๆผ่านทางบัสข้อมูลภายใน 8255 เอง สำหรับบล็อกการทำงานทางด้านซ้าย ที่มีชื่อว่า Data bus buffer และ read/write control logic ทำหน้าที่เชื่อมต่อระหว่างระบบบัสของไมโครคอนโทรลเลอร์กับ 8255 เพื่อรับหรือส่งข้อมูลระหว่างกันตามระดับลอจิกของขาสัญญาณ RD\ และ WR\ ตามลำดับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.17 แสดงแผนภาพแบบบล็อคภายในและขาสัญญาณของไอซีเบอร์ 8255

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.10 การจำแนกกลุ่มของพอร์ท 8255

ในบรรดาพอร์ททั้งสามของ 8255 คือ พอร์ท A,B และ C โดยพื้นฐานนั้น ล้วนเป็นพอร์ทแบบขนานที่ประกอบด้วยสัญญาณ 8 เส้นซึ่งแต่ละเส้นจะแทนบิตของข้อมูลพอร์ท ซึ่งอาจจะกล่าวได้อีกลักษณะว่าเป็นพอร์ทแบบ 8 บิต นอกจากนี้ยังสามารถอ้างถึงแต่ละบิตของเส้นสัญญาณพอร์ทนี้ได้โดยอิสระ อย่างไรก็ตาม 8255ได้จัดกลุ่มของพอร์ทเหล่านี้ออกเป็นสองกลุ่ม คือ กลุ่ม Aและ B เพื่อประโยชน์ในการกำหนดรูปแบบการทำงานของพอร์ท ดังตารางต่อไปนี้

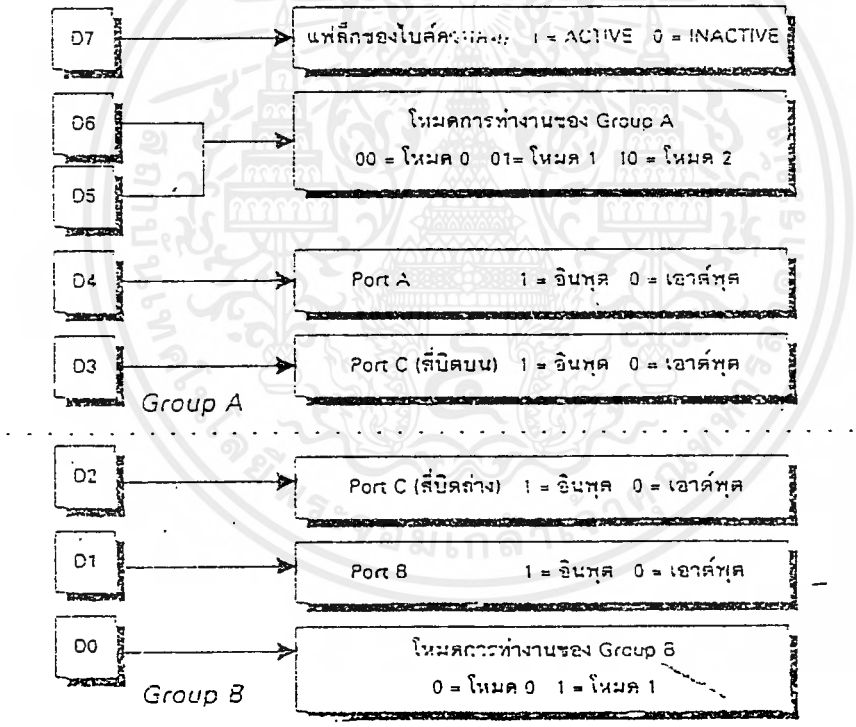
ชื่อกลุ่ม	ลักษณะ
GROUP A	พอร์ท A จำนวน 8 บิต (ทุกบิตของพอร์ท) พอร์ท C จำนวน 4 บิต (เฉพาะ 4 บิตบนของพอร์ท)
GROUP B	พอร์ท B จำนวน 8 บิต (ทุกบิตของพอร์ท) พอร์ท C จำนวน 4 บิต (เฉพาะ 4 บิตล่างของพอร์ท)

จากตารางข้างต้นจะเห็นว่า จำนวนเส้นสัญญาณทั้งหมดของพอร์ท C (PC0-PC7) ได้ถูกแยกออกเป็นกลุ่ม คือ กลุ่มของ 4 บิตล่าง (Lower nibble) จากPC0-PC3 และกลุ่มของ 4บิตบน (Upper nibble)จาก PC4-PC7ดังนั้น GROUP A and GROUP B ของ 8255 จึงมีจำนวนบิตในแต่ละกลุ่มเป็นจำนวนถึง 12บิต

สัญญาณ	ความหมาย
D0 -D7	กลุ่มของเส้นสัญญาณข้อมูลของ 8255 เมื่อมีการเขียน หรือ อ่าน
CS\	สัญญาณเลือกอุปกรณ์ เมื่อขาของสัญญาณนี้เป็นระดับลอจิกต่ำ ซีพียูก็สามารถ เขียนหรืออ่าน ข้อมูลจาก 8255 ได้
RD\	สัญญาณบอกสถานะต้องการอ่านข้อมูลจากรีจิสเตอร์ของ 8255
WR\	สัญญาณบอกสถานะต้องการเขียนข้อมูลให้กับรี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	จิสเตอร์ของ 8255
A0 -A1	สัญญาณระบุตำแหน่งรีจิสเตอร์ภายใน 8255 ที่ต้องการ
RESET	สัญญาณการรีเซตวงจรทำงานภายใน 8255 เพื่อเริ่มต้นใหม่
PA0 - PA7	กลุ่มของสัญญาณ 8 เส้น เมื่อทำการติดต่อกับพอร์ท A
PB0 - PB7	กลุ่มของสัญญาณ 8 เส้น เมื่อทำการติดต่อกับพอร์ท B
PC0 - PC7	กลุ่มของสัญญาณ 8 เส้น เมื่อทำการติดต่อกับพอร์ท C



รูปที่ 3.18 ความหมายของบิตภายในไบต์ข้อมูลควบคุมสำหรับ8255

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.11 รูปแบบคำสั่งเพื่อกำหนดการทำงานของ 8255

การกำหนดให้พอร์ททั้งสามของ 8255 ทำงานในลักษณะต่างๆกันหรือที่เรียกว่า โหมดการทำงาน(MODE) จะเริ่มด้วยการส่งค่าข้อมูลไบต์หนึ่งให้กับรีจิสเตอร์ควบคุมการทำงานภายใน 8255 ข้อมูลนี้จะเรียกว่าไบต์ข้อมูลควบคุม (Control word) โดยแต่ละบิตของข้อมูลนี้จะมีความหมายที่จะระบุถึงความต้องการต่างๆ ไปดังแสดงในรูปที่ 3 การส่งข้อมูลไบต์นี้จะต้องเริ่มต้นเป็นลำดับแรกก่อนที่จะได้มีการดำเนินกรใดกับ 8255 ทั้งสิ้น

ตามความหมายของบิตภายในตารางของรูปที่ 3.18 จะเห็นได้ว่าการเลือกให้พอร์ทใดทำหน้าที่เป็นพอร์ทอินพุทก็เพียงแต่กำหนดค่าข้อมูล 1 ให้กับบิตที่เกี่ยวข้องกับพอร์ทนั้น หรือกรณีตรงข้ามสำหรับการเอาท์พุทก็เพียงกำหนดค่าข้อมูล 0 เท่านั้น อย่างไรก็ตามการกำหนดให้ไบต์ข้อมูลควบคุมนี้มีผลอย่างถูกต้อง ก็จะต้องทำการกำหนดให้บิต D7 มีค่าเป็น 1 เสมอ สำหรับบิตที่บอกถึงโหมดการทำงาน (บิต D6 - D5 และ D2) นั้นจะได้กล่าวรายละเอียดต่อไป

3.12 การเชื่อมต่อ 8255 กับ 8051

เมื่อพิจารณาแผนภาพของ 8255 จะเห็นว่ามีขาสัญญาณแอดเดรสจำนวน 2 เส้น คือ A0 และ A1 ทำให้ตำแหน่งของแอดเดรสที่จะอ้างถึงได้มีค่าเป็น 2^2 หรือเท่ากับ 4 ตำแหน่ง ซึ่งแต่ละตำแหน่งจะมีความหมายถึงการระบุรีจิสเตอร์หรือพอร์ทภายใน 8255 ดังแสดงได้ดังนี้

A1	A0	ชื่อของรีจิสเตอร์
0	0	พอร์ท A
0	1	พอร์ท B
1	0	พอร์ท C
1	1	รีจิสเตอร์ควบคุม

เมื่อพิจารณาค่าของแอดเดรสเหล่านี้ร่วมกับระดับลอจิกของขาสัญญาณ RD\ และ WR\ จะเป็นการอ่านหรือเขียนข้อมูลทางขาสัญญาณ D0-D7 ให้กับรีจิสเตอร์นั้นตามลำดับ ดังตารางต่อไปนี้

RD\	WR\	A1	A0	ความหมาย
0	1	0	0	ส่ง(หรือเขียนข้อมูลให้กับพอร์ท A

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1	0	0	0	รับ(หรืออ่าน)ข้อมู จากพอร์ท A
0	1	0	0	ส่ง(หรือเขียน ข้อมูลให้กับพอร์ B
1	0	0	1	รับ(หรืออ่าน)ข้อมู จากพอร์ท B
0	1	1	0	ส่ง(หรือเขียน ข้อมูลให้กับพอร์ C
1	0	1	0	รับ(หรืออ่าน)ข้อมู จากพอร์ท C
0	1	1	1	ส่ง(หรือเขียน)ให้ กับรีจิสเตอร์คว คุม
1	0	1	1	เป็นสภาวะที่ไม่ถู ต้อง

ดังนั้นโดยทั่วไปจึงมักจะกำหนดให้แอดเดรสของ 8255 ทั้งสี่ตำแหน่ง อยู่ในแอดเดรสช่วงใดช่วงหนึ่งของระบบ เช่น 10 h, 11h, 12h และ 13h โดยขาสัญญาณแอดเดรสที่นอกเหนือไปจาก A0 และ A1 นำมาเข้ายังตัวถอดรหัสแอดเดรส เพื่อสร้างสัญญาณเลือกอุปกรณ์ (CS\) ในช่วงแอดเดรสที่ต้องการ ให้ดูตัวอย่างในรูป 3.19 สัญญาณ CS\ นี้จะเป็นสภาวะลอจิกต่ำก็ต่อเมื่อค่าในบัสแอดเดรส A2 - A7 มีค่าเท่ากับ 0000100xx (xx ใช้เพื่อระบุถึงรีจีสเตอร์ภายใน 8255 เพื่อทำการอ่านหรือเขียนข้อมูล) ดังนั้นจากวงจรนี้แอดเดรสของรีจีสเตอร์ภายใน 8255 จะมีค่าตามตารางต่อไปนี้

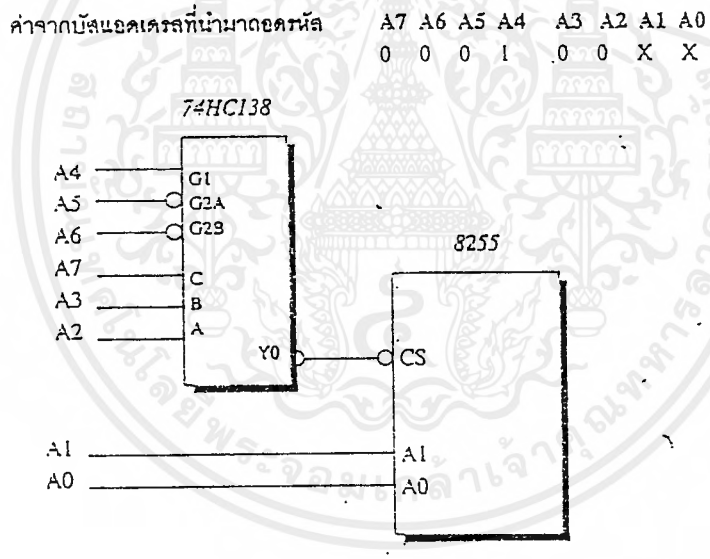
ตำแหน่งแอดเดรส	ความหมาย
10h	พอร์ต A
11h	พอร์ต B
12h	พอร์ต C

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

13h	รีจีสเตอร์ควบคุม

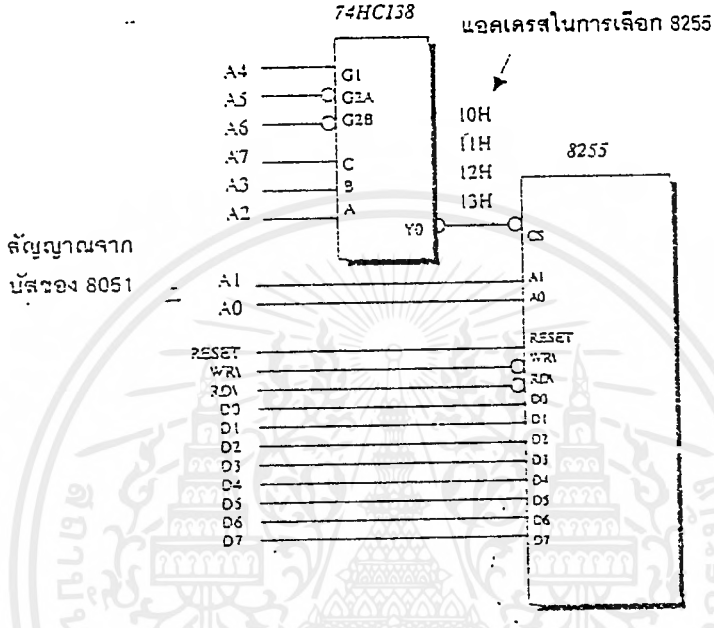
ขาสัญญาณควบคุมอื่นๆ คือ RD\ และ WR\ มักจะเชื่อมต่อเข้ากับขาสัญญาณชื่อเดียวกันของ 8051 ได้โดยตรง ทำให้แอดเดรสพอร์ตของ 8255 อยู่ในพื้นที่ของหน่วยความจำข้อมูลของ 8051 สำหรับขาสัญญาณรีเซ็ตของ 8255 ซึ่งจะมีผลทำให้เกิดการรีเซ็ต หรือ เริ่มสภาวะการทำงานใหม่เมื่อระดับของขาสัญญาณเป็นลอจิกสูง ดังนั้นหากว่าจะใช้สัญญาณการรีเซ็ตเดียวกับของ 8051 เพื่อที่จะรีเซ็ต 8255 ด้วยก็สามารถต่อได้โดยตรง

ส่วนขาสัญญาณ D0 - D7 ก็สามารถนำไปเชื่อมต่อโดยตรงเข้ากับบัสของ 8051 ได้เช่นกัน ขอให้ดูวงจรทำงานแสดงการเชื่อมต่อระหว่าง 8255 และ 8051 ในรูปที่ 3.20 ซึ่งในที่นี้สมมุติว่าไม่จำเป็นต้องมีการใช้วงจรหรือไอซี บัฟเฟอร์ ขับสัญญาณบัสข้อมูล



รูปที่ 3.19 แผนภาพแสดงการสร้างสัญญาณเลือกอุปกรณ์(CS\) ให้กับ8255โดยการถอดรหัสจากบัสแอดเดรส A2-A7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 320 แผนภาพวงจรแสดงการเชื่อมต่อระหว่าง 8255 กับ 8051

3.13 การทำงานโหมด 0 ของ 8255

เมื่อ 8255 ได้รับการกำหนดให้ทำงานในโหมดนี้ จะทำให้พอร์ตต่างๆ มีหน้าที่เป็นพอร์ตอินพุต หรือเอาต์พุตได้เพียงลักษณะเดียวเท่านั้น การเริ่มต้นจะทำโดยการส่งไบต์ข้อมูลควบคุมให้กับรีจิสเตอร์ควบคุม (ขอให้ดูความหมายของบิต จากรูปที่ 3) ต่อไปจะได้แสดงให้เห็นถึงรูปแบบการกำหนดบิต เมื่อต้องการให้พอร์ต A, B และ C ทำหน้าที่เป็นพอร์ตเอาต์พุตทั้งหมด ดังตารางต่อไปนี้

ตำแหน่งบิต	ค่าข้อมูล	ความหมาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

D7	1	ระบุให้ทราบว่าเป็นไบต์ข้อมูลควบคุม
D6 และ D5	00	กำหนดโหมดการทำงานให้กับพอร์ต A เป็นโหมด 0
D4	0	ระบุว่าพอร์ต A เป็นการเอาต์พุตข้อมูล
D3	0	กำหนดให้เส้นสัญญาณสี่บิตบนของพอร์ต C เป็นการเอาต์พุตข้อมูล
D0	0	กำหนดโหมดการทำงานให้กับพอร์ต B เป็นโหมด 0
D1	0	ระบุว่าพอร์ต B เป็นการเอาต์พุตข้อมูล
D2	0	กำหนดให้เส้นสัญญาณสี่บิตล่างของพอร์ต C เป็นการเอาต์พุตข้อมูล

ค่าของไบต์ข้อมูลควบคุมนี้จะต้องส่ง (หรือ เขียน) ให้กับรีจีสเตอร์ควบคุม ซึ่งหากใช้วงจรตามรูปที่ 5
จะเป็นแอดเดรส 13h และสามารถแสดงคำสั่งได้ดังนี้

```
MOV   DPTR,#4003H        ;SET THE CONTROL REGISTER ADDRESS
MOVX @DPTR,#80H          ;OUTPUT THE CONTROL WORD TO 8255
```

ตามตัวอย่างข้างต้น เนื่องจากแอดเดรสมีค่าไม่เกิน 256 (0ffh) จึงสามารถอ้างถึงตำแหน่งแอดเดรสของรีจีสเตอร์ควบคุมของ 8255 ผ่านทางรีจีสเตอร์ R0ได้ แต่ในบางกรณีค่าของแอดเดรสนี้อยู่ในช่วงที่มีค่าเกินกว่านี้ เช่น แอดเดรส 4000h ถึง 4003h ก็จะต้องใช้การอ้างอิงถึงโดยอ้อมผ่านทางรีจีสเตอร์ DPTR แทนดังนี้

```
MOV   DPTR,#4003H        ;SET THE CONTROL REGISTER ADDRESS
MOVX @DPTR,#80H          ;OUTPUT THE CONTROL WORD TO 8255
```

ภายหลังจาก 8255 ได้รับการโปรแกรมจากค่าของไบต์ข้อมูลควบคุมนี้แล้ว ก็สามารถที่จะใช้งานพอร์ตทั้งหมดในฐานะพอร์ตเอาต์พุตเพื่อส่งออกข้อมูลได้ตามต้องการ ตัวอย่างเช่น หากต้องการส่งข้อมูลค่า 23h ออกทางพอร์ต A ค่าข้อมูล 41h ออกทางพอร์ต B และค่าข้อมูล 73h ออกทางพอร์ต C จะมีรูปแบบคำสั่งดังนี้

```
MOV   R0,#10H                ;SET UP PORT A ADDRESS
MOV   @R0,23H            ;OUTPUT DATA TO 8255
MOV   R0,#11H                ;SET UP PORT B ADDRESS
MOV   @R0,41H            OUTPUT DATA TO 8255
MOV   R0,#12H                ;SET UP PORT C ADDRESS
MOV   @R0,73H            ;OUTPUT DATA TO 8255
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กรณีต่อไป หากต้องการให้พอร์ตของ 8255 มีทั้งพอร์ตอินพุตและเอาต์พุต ตัวอย่างเช่น ต้องการให้พอร์ต A และ C เป็นพอร์ตเอาต์พุท สำหรับพอร์ต B เป็นพอร์ตอินพุต ค่าของไบต์ควบคุมจะเป็นดังนี้

ตำแหน่งบิต	D7	D6	D5	D4	D3	D2	D1	D0
ค่าของบิต	1	0	0	0	0	0	1	0

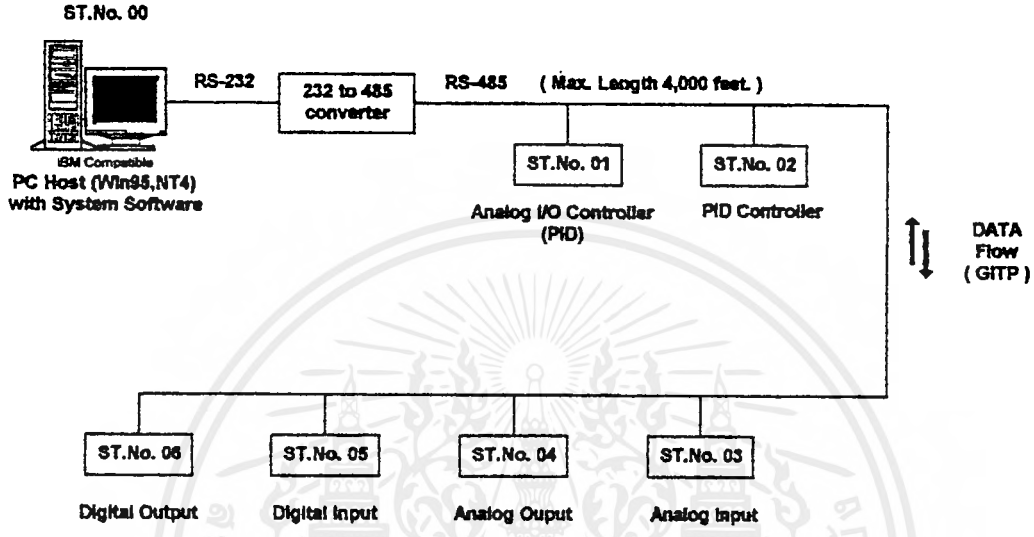
ตัวอย่างคำสั่งเพื่อการอ่านข้อมูลเข้ามาทางพอร์ต B

```
MOV   R0,#11H                ;SET UP PORT B ADDRESS
MOV   A,@R0                  ;READ DATA FROM PORT B
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

Small Scale DCS



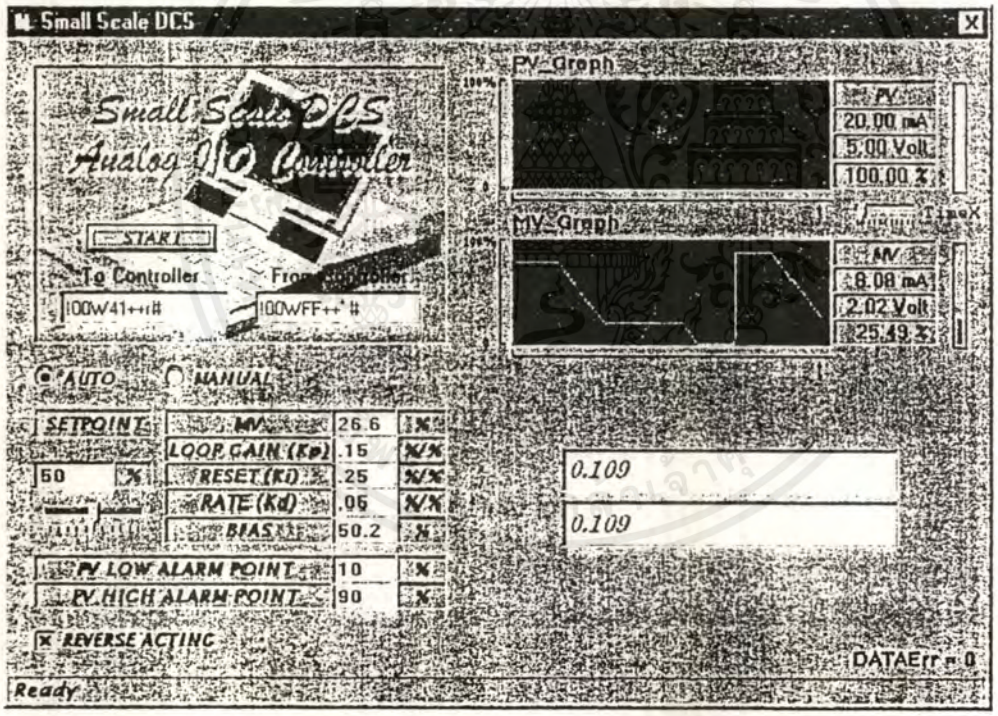
4.1 ลักษณะการทำงานของระบบที่สร้างขึ้น

โครงสร้างของโครงงานนี้ประกอบด้วย 8 สถานีงาน คือ Station number 00 -06 และ 232 to 485 converter รายระเอียดดังนี้ คือ

1. ST.No. 00 เป็น เครื่องไมโครคอมพิวเตอร์ IBM Compatible ที่ใช้ระบบปฏิบัติการ Windows95 หรือ WindowsNT4 และ ใช้โปรแกรม System Software ที่ เขียนขึ้นโดยเฉพาะ สำหรับ Small Scale DCS ชุดนี้ โปรแกรม Software นี้ เขียนโดยใช้ Visual Basic 4.0 .ให้ทำงานในโหมด 32 บิต เท่านั้น ST.No. 00 นี้จะทำหน้าที่เป็นตัวควบคุมระบบ ทั้งหมดโดย ทำหน้าที่ เป็น Master Station (Host) การติดต่อสื่อสารระหว่าง Station ต่างๆ ใช้โปรโตคอล General Instrument Protocol (GITP)

2. ST.No. 01 คือ Analog Input/Output Controller ซึ่งทำงานเป็น PID Controller แบบ Direct Digital Control คือ การคำนวณ สมการ PID จะทำโดย Software ที่ Computer (ST.No.00) ทั้งหมด โดย Process Variable และ Manipulate Variable จะส่งผ่านทาง Analog Input/Output Controller

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



System Software

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. ST.No. 02 คือ PID Controller โดยจะต่างกับ ST.No. 01 คือ การคำนวณ สมการ PID จะทำโดย Software ที่ Controller เองโดยตรง แล้วจะทำการส่งข้อมูล Process Variable และ Manipulate Variable ไปแสดงผลที่ Host Computer (ST.No. 00)

4. ST.No. 03 Analog Input ทำหน้าที่ รับสัญญาน Analog Input (4-20mA) แล้วจะทำการส่งข้อมูล ไปให้ PC Host (ST.No. 00) ทำการประมวลผลตามโปรแกรมที่ต้องการ

5. ST.No. 04 Analog Output ทำหน้าที่ รับค่าจาก PC Host (ST.No. 00) มาออกเป็น Output Analog (4-20mA) ตามโปรแกรมที่กำหนดไว้ใน System Software

6. ST.No. 05 Digital Input ทำหน้าที่ รับสัญญาน Input แบบ Digital 8 bit ส่งให้ PC Host (ST.No. 00) นำไปทำการประมวลผลตามโปรแกรมที่ต้องการ ต่อไป

7. ST.No. 06 Digital Output ทำหน้าที่ รับค่าจาก PC Host (ST.No. 00) มาออกเป็น Output แบบ Digital ตามเงื่อนไขโปรแกรมที่กำหนดไว้ใน System Software

8. RS-232 to RS-485 Converter เป็น ตัวแปลงสัญญาน ระหว่าง มาตรฐาน RS-232 และ RS-485 เพราะ พอร์ตอนุกรมในคอมพิวเตอร์ทั่วไป จะใช้มาตรฐาน RS-232 ซึ่งไม่สามารถใช้สายสัญญานยาวได้ และ 1 พอร์ต สามารถต่อ อุปกรณ์ได้เพียง 1 ตัว เท่านั้น เมื่อจะสร้างโครงงานนี้ ให้มีอุปกรณ์ หลายตัว และสามารถ ใช้สายสัญญานยาวได้ จึงต้องแปลง ให้เป็นมาตรฐาน RS-485 ซึ่งสามารถใช้ สายได้ยาวสูงสุด ถึง 4,000 ฟุต และ สามารถต่ออุปกรณ์ บนพอร์ตเดียวได้สูงสุด ถึง 32 ตัว

การทำงานของระบบ PC Host (ST.No. 00) จะส่ง DATA Packet ออกไปที่พอร์ต RS-232 ผ่าน RS-232 to RS-485 Converter เป็นมาตรฐาน RS-485 ไปยัง Module ต่างๆ เพื่อให้ตอบสนองต่อคำสั่ง และส่ง Data Packet กลับเข้ามา ยัง PC Host (ST.No. 00) ที่ PC Host (ST.No. 00) นี้จะมีการประมวลผลต่อข้อมูลที่รับเข้ามา และส่งกลับออกไปอีก การทำงานจะวน Loop ตามโปรแกรมที่ได้ทำเอาไว้ การแยก Data packet ของแต่ละ Module จะอาศัย ST.No. เป็นตัวแยก ว่าData packet ที่ส่งมาจาก PC Host (ST.No. 00) เป็นของ Module ใด โดย Module อื่นๆ จะไม่ตอบสนองต่อ Data packet ที่มี ST.No. ไม่ตรงกับตัวมัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 General Instrument Protocol (GITP)

การกำหนดโปรโตคอล (Protocol) ในการติดต่อสื่อสารระหว่าง คอมพิวเตอร์หลัก (Host Computer) กับ Controller Module ต่างๆ เราได้ออกแบบ Protocol ขึ้นมาใช้เองโดยมีข้อกำหนด คือ ต้องง่ายไม่ซับซ้อน ,ขนาดของ Packet ข้อมูลเหมาะสมไม่ยาวจนเกินไป และ มีความยืดหยุ่นสูง สามารถ ประยุกต์ใช้ ได้อย่างกว้างขวาง โดยใช้ชื่อว่า General Instrument Protocol (GITP)

ตารางข้อกำหนดของ GITP

Packet byte Number	Packet type	Character available
0	Start	!
1	ST.No	0-9
2	ST.No	0-9
3	General Command & Status	A-Z
4	DATA	0-F
5	DATA	0-F
6	General purpose	All
7	General purpose	All
8	Check Sum	All
9	End	#

GITP มีขนาดของ DATA Packet เท่ากับ 10 Byte แบ่งออกเป็น 7 ส่วน แต่ละ Byte เก็บข้อมูล อักขระ ASCII 1 ตัว มีข้อกำหนดต่างๆ ดังตาราง แยกอธิบายได้ดังนี้

ส่วนที่ 1 "START" ประกอบด้วย ข้อมูล 1 Byte คือ Byte 0 เป็นส่วนที่ทำหน้าที่บอกสถานะเริ่มต้นของ Packet อักขระที่ใช้ได้มีเพียงตัวเดียว คือ "!"

ส่วนที่ 2 "ST.No." ประกอบด้วย ข้อมูล 2 Byte คือ Byte 1,2 เป็นส่วนที่ทำหน้าที่บอกหมายเลขของอุปกรณ์ที่ติดต่อ อักขระที่ใช้ได้คือตัวเลข "0-9"

ส่วนที่ 3 "General Command & Status" ประกอบด้วย ข้อมูล 1 Byte คือ Byte 3 เป็นส่วนที่กำหนดคำสั่ง และหรือ บอกสถานะ ของอุปกรณ์ อักขระที่ใช้ได้ คือ "A-Z"

ส่วนที่ 4 "DATA" ประกอบด้วย ข้อมูล 2 Byte คือ Byte 4,5 เป็นส่วนที่ทำหน้าที่ส่งผ่านค่า Input,Output ของอุปกรณ์ อักขระที่ใช้ได้คือ "0-F"

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนที่ 5 "General purpose" ประกอบด้วย ข้อมูล 2 Byte คือ Byte 6,7 เป็นส่วนที่สามารถเขียนโปรแกรมนำไปใช้อย่างไรก็ได้ สามารถใช้อักขระได้ทุกตัว

ส่วนที่ 6 "Check Sum" ประกอบด้วย ข้อมูล 1 Byte คือ Byte 8 เป็นส่วนที่ใช้สำหรับตรวจสอบความถูกต้องของ DATA Packet สร้างจากการนำ ข้อมูล Byte 1-7 ใน Packet นั้นๆ มาบวกกันโดยไม่คิดตัวทด สามารถใช้อักขระได้ทุกตัว

ส่วนที่ 7 "END" ประกอบด้วย ข้อมูล 1 Byte คือ Byte 9 เป็นส่วนที่ทำหน้าที่บอกการสิ้นสุดของ Packet อักขระที่ใช้ได้มีเพียงตัวเดียว คือ "#"

General Command & Status (GSC) ที่กำหนดใช้แล้ว

"W"	หมายถึง	การทำงาน เขียนข้อมูล ไปที่ อุปกรณ์เป้าหมาย
"E"	หมายถึง	ข้อมูลที่ โมดูลต่างๆ ได้รับจาก Computer (ST.No.00) ChkSum ไม่ถูกต้อง โดย โมดูลต่างๆ จะส่ง GSC = "E" กลับมาที่ Computer ในรอบถัดไป
"B"	หมายถึง	ไหมด ตรวจสอบโดยจะแสดงข้อมูล ที่ได้รับแต่ละไบต์ที่ LCD และแสดงค่า PV และ MV เป็นเลขฐาน 16 ใช้สำหรับ ST.No. 01 (Analog Input/Output Controller)
"R"	หมายถึง	หมุนเสต็ปมอเตอร์ ไปทางขวา ใช้สำหรับ ST.No. 06 (Digital Output)
"L"	หมายถึง	หมุนเสต็ปมอเตอร์ ไปทางซ้าย ใช้สำหรับ ST.No. 06 (Digital Output)
"S"	หมายถึง	หยุดหมุนเสต็ปมอเตอร์ ใช้สำหรับ ST.No. 06 (Digital Output)
"M"	หมายถึง	Manual MODE (SP) ใช้สำหรับ ST.No. 02 (PID Unit)

ตัวอย่าง Packet ของ GITP

Byte	0	1	2	3	4	5	6	7	8	9
	!	0	6	W	0	0	+	+	s	#
	START	ST.No.		Cmd,Status	DATA		General purpose		ChkSum	END

จากตัวอย่าง หมายถึง Packet นี้ เป็นของ อุปกรณ์หมายเลข 06 (ST.No. 06) Cmd,Status = "W" หมายถึงการทำงาน เขียนข้อมูล ไปที่ อุปกรณ์เป้าหมาย DATA ที่ต้องการส่งคือ 00H ไม่มีการใช้งาน Byte 6,7 (General purpose) ค่า ChkSum แปลงเป็น ASCII ได้ อักขระ "s"

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปัญหาที่เกิดขึ้นในการเขียนโปรแกรม System Software

1. ข้อมูลในการเขียนโปรแกรมการติดต่อสื่อสาร อนุกรม RS-485 มีน้อยมาก โดยเฉพาะในภาษาที่ใช้ คือ Visual Basic 4.0

2. ความละเอียดฐานเวลาของ IBM Compatible นั้นต่ำ โดยใน 1 วินาที จะมีการ UPDATE กับวงจรนาฬิกาภายในเครื่อง เพียง 18 ครั้ง เท่านั้น ทำให้ได้ความละเอียดประมาน 55 ms ต่อ การ UPDATE (Tick) 1 ครั้ง ในการจะใช้ความละเอียดสูงกว่านี้ จะต้องเขียนรูทีน พิเศษขึ้นมา

3. ความเร็วของเครื่องถ้าต่ำเกินไปจะทำให้ทำการรันโปรแกรม ใน 1 รอบ ไม่ทันทำให้การคำนวณของโปรแกรมผิดพลาด

4. ในการติดต่ออนุกรม RS-485 ที่เป็นแบบ Half Duplex จะแตกต่าง จาก RS-232 ซึ่งเป็น Full Duplex ในการเขียนโปรแกรมจึงต้องเกี่ยวข้องกับ Flow Control ซึ่งยาก และ ไม่ได้ผลนัก โดยในบางครั้ง ก็จะทำงาน ขาดๆ หายๆ และ ทำให้ Computer Hang

5. โปรแกรมเอสเซสบลี ในโมดูลต่างๆ จะมีการติดต่อสื่อสาร โดยใช้ GITP อยู่ด้วย และจะต้องรวมกับ โปรแกรมเอสเซสบลี ที่ใช้ในโมดูลนั้นๆ ซึ่ง โปรแกรมที่ไม่ได้เขียนคนเดียว เวลาที่จะนำมารวมกันให้ทำงานได้ จะมีปัญหามากพอสมควร

แนวทางในการพัฒนาเพิ่มเติม

1. เพิ่มเสถียรภาพของระบบ ติดต่ออนุกรม RS-485 เช่น เพิ่ม Isolate (OptoCoupler devices) เพิ่มสายกราวน์ อ้างอิง ถึงทุกโมดูล

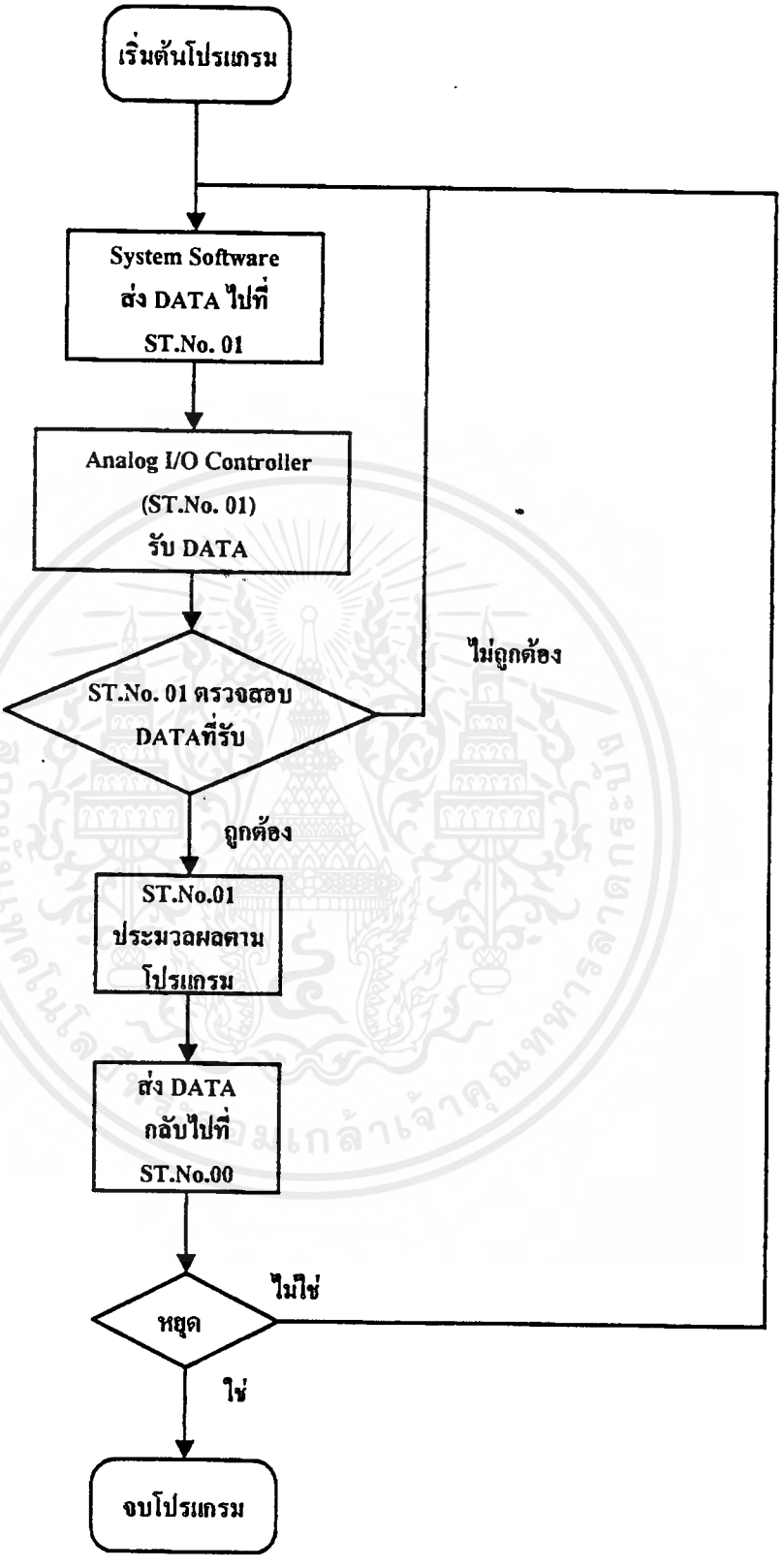
2. เพิ่มความเร็ว รอบการทำงาน และ อัตราบ๊อด

3. ส่วนแสดงกราฟ อาจพัฒนาให้มีการเก็บค่าเอาไว้ และสามารถเรียกกลับมาดูได้ เสมือนเป็น Recorder

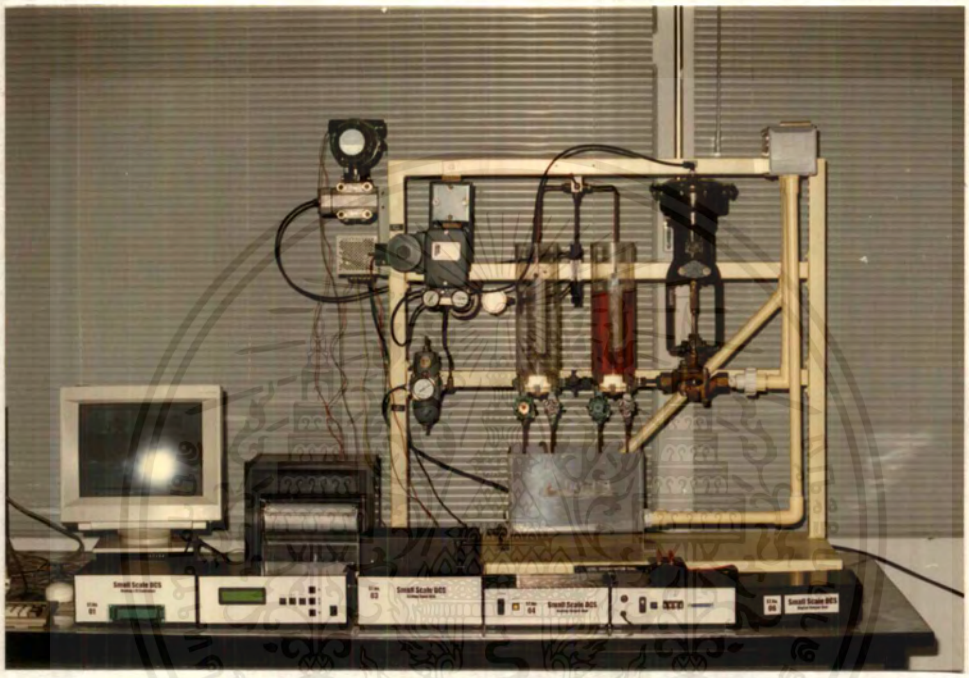
4. สร้างส่วนฐานเวลาของ System Software ให้มีความละเอียด และแม่นยำมากขึ้น โดยอาจใช้ ภาษาอื่นๆ เช่น C++ หรือ Assembly สร้างเป็นไฟล์ Dynamic-Link Library (DLL) แล้วเรียกใช้อีกครั้งหนึ่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

System Software



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปแสดง UNIT ต่างๆ ในระบบ SMALL SCALE DCS

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

PROCESS CONTROL

Process control คือการควบคุมหรือการกระทำใดๆ เพื่อให้เกิดสภาวะทางฟิสิกส์ ตามที่เราต้องการที่ตัว process นั้นๆ ซึ่งเป้าหมายของการควบคุม process ต่างๆ ในโรงงานก็คือ การรักษาปริมาณทางฟิสิกส์ เช่น อุณหภูมิ แรงดัน รักษาระดับของเหลว ตลอดจนอัตราการไหลของของไหล เป็นต้น ในการควบคุมรักษาปริมาณเหล่านี้ต้องการความเร็ว ความแม่นยำ และความเชื่อถือ เพื่อประสิทธิภาพในการผลิตของโรงงานสมัยใหม่ ดังนั้นจึงมีการคิดค้นวิธีการควบคุมในแบบอัตโนมัติขึ้นมา และพยายามพัฒนาเพื่อให้เกิดประสิทธิภาพในการควบคุมมากที่สุด ในบทนี้เราจะกล่าวถึงการควบคุม process และองค์ประกอบสำคัญในการควบคุมพร้อมทั้งตัวแปรและสัญญาณต่างๆ เพื่อเป็นพื้นฐานไปสู่บทต่อๆ ไป

5.1การควบคุม PROCESS

ในการควบคุม process แบบอัตโนมัติ จะสามารถแสดงได้ดัง block diagram ดังแสดงในรูปที่ 5.1นี้ ซึ่งจะประกอบไปด้วยองค์ประกอบต่างๆ และนิยามจำกัดความดังนี้

1.โปรเซส (PROCESS) หมายถึง ขบวนการทางฟิสิกส์ที่เราต้องการควบคุมให้มีสภาวะตามที่เราต้องการ ขณะที่สภาวะการทำงาน หรือสภาวะแวดล้อมอาจจะเปลี่ยนแปลงตลอดเวลา ตัวอย่างของ process ก็คือ ขบวนการรักษาระดับน้ำในทั้งในถังเก็บให้คงที่ หรือการรักษาอุณหภูมิในเตาอบให้มีค่าตามต้องการ หรือการรักษาอัตราไหลให้มีค่าคงที่ เป็นต้น

2. ตัวแปรโปรเซส (PROCESS VARIABLE) หมายถึง ตัวแปรทางฟิสิกส์ซึ่งแสดงสภาวะทางฟิสิกส์ของ process เช่น ระดับน้ำเป็นเมตร อุณหภูมิเป็นองศา เป็นต้น ในการควบคุมเราจะวัดค่าตัวแปร process นี้มาแล้ว แปลงเป็นสัญญาณมาตรฐานก่อนที่จะนำมาวิเคราะห์เพื่อควบคุมต่อไป

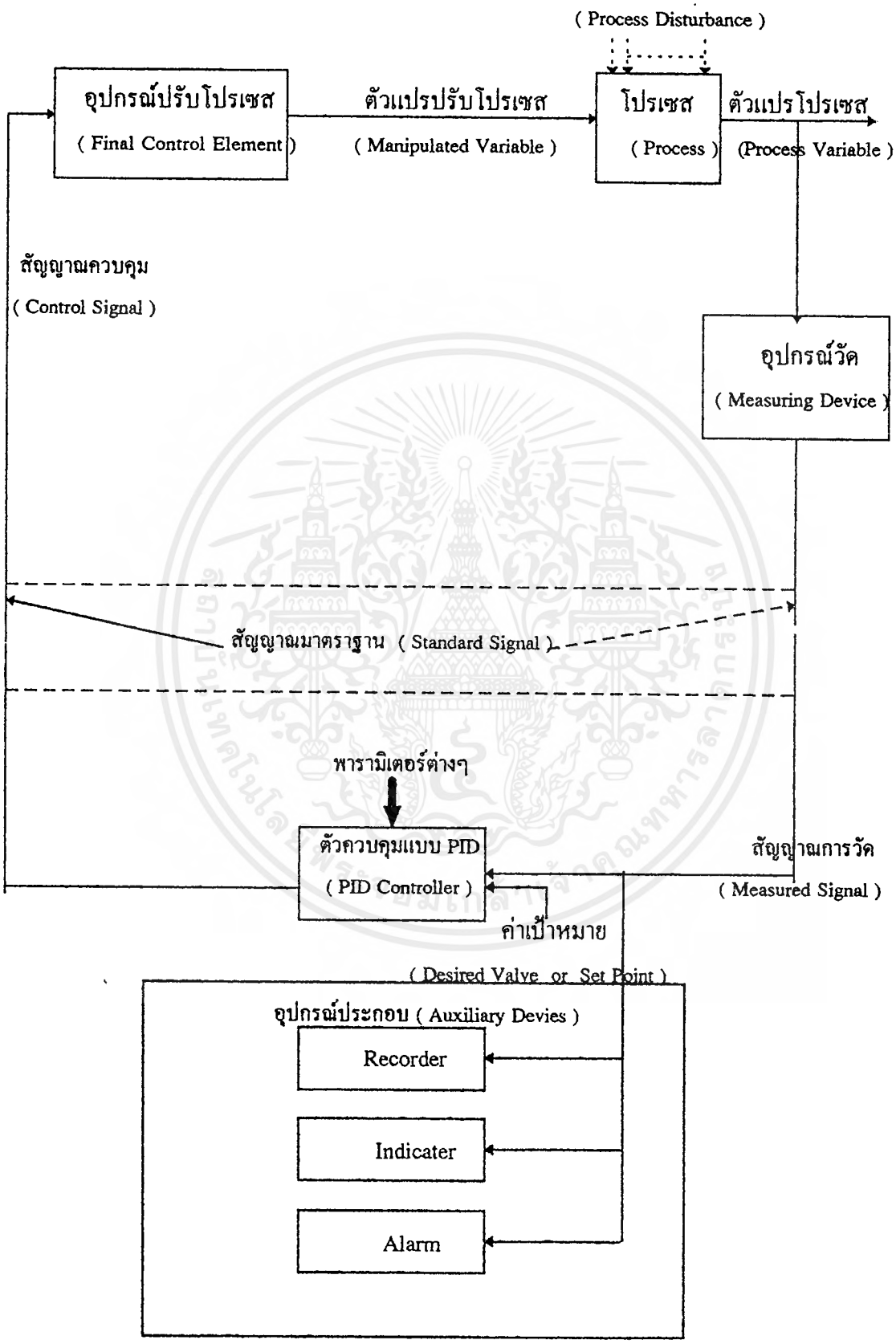
3.ตัวแปรปรับโปรเซส (MANIPULATED VARIABLE) หมายถึง ตัวแปรที่จะใช้สำหรับเปลี่ยนแปลงค่าตัวแปร process ให้ได้ดังต้องการซึ่งได้แก่ อัตราการไหลของน้ำ เป็นต้น

4.อุปกรณ์การวัด (MEASURING DEVICE) หมายถึง อุปกรณ์ที่เราใช้วัดค่าตัวแปร process (PV) ออกมาเป็นสัญญาณมาตราฐาน หรือกล่าวได้ว่าเป็นอุปกรณ์ที่ให้สัญญาณมาตราฐานออกมา มีขนาดสัมพันธ์กับขนาดของตัวแปรทางฟิสิกส์ของสิ่งที่เราต้องการวัด โดยทั่วไปต้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ให้ความสัมพันธ์ดังกล่าว เช่น สมการเชิงเส้น ตัวอย่างของอุปกรณ์วัดก็คือ เทอร์โมคับเปิ้ลวัดอุณหภูมิ เป็นต้น

รูปที่ 5.1 Block diagram ระบบควบคุมแบบป้อนกลับ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.ตัวควบคุม (CONTROLLER) หมายถึง ตัวที่ทำหน้าที่ออกคำสั่ง หรือกำเนิดสัญญาณควบคุม (Control Signal) ตามกฎเกณฑ์ที่กำหนดไว้ล่วงหน้า คำสั่งหรือสัญญาณควบคุมนี้อาจจะเป็นฟังก์ชั่นกับเวลา หรือฟังก์ชั่นกับสัญญาณขาเข้าที่ได้รับจากอุปกรณ์วัด ในการควบคุมแบบอัตโนมัติ ตัวควบคุมได้แก่ เครื่องควบคุมอัตโนมัติ (Automatic controller) ซึ่งจะรับสัญญาณจากอุปกรณ์วัด คำนวนหาสัญญาณควบคุมที่เหมาะสมตามกฎเกณฑ์ที่ได้รับการกำหนดไว้ล่วงหน้าแล้วส่งสัญญาณไปที่อุปกรณ์ปรับ process สำหรับกฎเกณฑ์การควบคุมนี้จะถูกกำหนดไว้ได้หลายรูปแบบด้วยกัน แบบหนึ่งที่นิยมกันมากก็คือการควบคุมชนิด PID ซึ่งเราจะศึกษากันในบทต่อไป

6. อุปกรณ์ปรับโปรเซส (FINAL CONTROL ELEMENT) หมายถึง อุปกรณ์ที่ทำหน้าที่ปรับสภาวะของโปรเซส ด้วยการเปลี่ยนแปลงค่าตัวปรับโปรเซส (Manipulated variable) ตามคำสั่งหรือสัญญาณควบคุมที่ได้รับ ตัวอย่างของอุปกรณ์ชนิดนี้คือ วาล์ว

เมื่อทราบถึงความหมายของสัญญาณและองค์ประกอบต่างๆในการควบคุมโปรเซสแล้ว เราจะสามารถพิจารณาการทำงานควบคุมดัง Block diagram ในรูปที่ ได้ดังนี้คือ

ความต้องการของเราก็คือ รักษาสภาวะทางฟิสิกส์ที่เราต้องการที่โปรเซสนั้น ให้มีค่าตามค่าเป้าหมาย ที่เรากำหนดไว้ แต่ process อาจจะได้รับการรบกวนจากสภาพแวดล้อมทำให้ค่าเป้าหมายผิดพลาดไป เราก็จะใช้อุปกรณ์วัดค่าตัวแปร process (PV) แล้วส่งมาเปรียบเทียบกับค่าเป้าหมาย (SP) ที่เรากำหนดไว้ที่ตัว Controller เพื่อให้ Controller ทำการคำนวนหาสัญญาณควบคุมที่เหมาะสมส่งไปยังอุปกรณ์ปรับ process เพื่อปรับค่าตัวแปร process ให้มีค่าเท่ากับค่าเป้าหมาย ซึ่งการทำงานที่กล่าวมานี้ก็คือหลักการควบคุมแบบป้อนกลับนั้นเอง สัญญาณรบกวน process (Process disturbance) ที่จะทำให้ค่าตัวแปร process เปลี่ยนแปลงไปนั้น เราจะจำแนกได้เป็น 2 ประเภทคือ

1.SUPPLY DISTURBANCE หมายถึง การเปลี่ยนแปลงของพลังงาน (วัตถุ) ขาเข้า เช่น การเปลี่ยนแปลงของอุณหภูมิ การเปลี่ยนแปลงความดันที่ไหลเข้าตัว process เป็นต้น

2.DEMAND DISTURBANCE หมายถึง การเปลี่ยนแปลงของพลังงาน (วัตถุ) ขาออก เช่น การเปลี่ยนแปลงของอุณหภูมิ อัตราการไหลของน้ำทางออกรวมไปถึงการสูญเสียพลังงานความร้อนที่ถังเก็บวัตถุของ process เป็นต้น

5.2 CONTROL SYSTEM PARAMETER

จากองค์ประกอบที่กล่าวมาแล้วในตอนต้น ทำให้เราสามารถสรุปถึงค่า parameter ที่มีผลต่อระบบควบคุมอัตโนมัติได้ดังนี้คือ

ERROR หรือเรียกอีกอย่างว่า DEVIATION เป็นตัวแปรคอนโทรลเลอร์ที่เกิดจากความบ่ายเบนของค่า process variable (PV) จากค่าเป้าหมายที่ตั้งไว้ (SP) แสดงได้ดังนี้คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

e = PV - SP

เมื่อ e = ERROR

PV = Process variable

SP = Setpoint

โดยทั่วไปแล้วในการควบคุมที่ดีเราจะไม่เพียงแต่หาค่าเบี่ยงเบนหรือ Error จากค่า PV และ SP เท่านั้น แต่จะคิดเทียบกับค่า full scale range เพราะมันจะให้ความหมายที่เป็นประโยชน์มากกว่าเพราะจะได้ค่า Error เป็นเปอร์เซ็นต์เทียบกับ full scale variable range สามารถเขียนได้เป็นสมการดังนี้

$$ep = \left[\frac{PV - SP}{PV_{max} - PV_{min}} \right]$$

เมื่อ Pvmax = maximum PV

Pvmin = minimum PV

ep = Error as percent of range from setpoint

ถ้า ep เป็นบวก แสดงว่าค่า PV มากกว่า Setpoint

ถ้า ep เป็นลบ แสดงว่าค่า PV น้อยกว่า Setpoint

สำหรับโครงงานนี้เราจะไม่ใช้ค่า ep ใช้เพียงค่า e ก็เพียงพอแล้ว

TIME LAG เมื่อเราส่งสัญญาณควบคุมให้แก่ process แล้ววัดดูการตอบสนองของตัวแปร process (PV) ในบาง process เราจะพบว่า PV จะมีค่าคงเดิมอยู่ชั่วขณะหนึ่ง แล้วจึงเกิดการเปลี่ยนแปลงตามสัญญาณควบคุม ลักษณะเช่นนี้แสดงให้เห็นว่า process มีความช้าในการตอบสนอง (Process time lag) ซึ่งเราจะพอแบ่งได้เป็นสองประเภทก็คือ

1.Deadtime คือช่วงเวลาที่นับจากขณะที่สัญญาณควบคุมเปลี่ยนแปลงไปจนกระทั่งสัญญาณวัดของตัวแปรโปรเซส (PV) เริ่มการเปลี่ยนแปลง จึงกล่าวได้ว่า Deadtime ทำให้การควบคุมยากขึ้น

2.Capacity lag คือช่วงเวลาที่สัญญาณวัด (PV) มีการตอบสนองกับสัญญาณควบคุมแล้ว แต่ยังไม่สามารถแปรค่า PV ให้เป็นตามนั้นได้ process ต้องใช้เวลาช่วงหนึ่งเพื่อค่อยๆ เปลี่ยนค่า PV ให้ถึงค่าที่ต้องการ

5.3 สัญญาณมาตราฐานที่ใช้ในการควบคุม

การเลือกใช้สัญญาณมาตราฐานใดๆ ในระบบควบคุมนั้น ขึ้นอยู่กับระบบ Instrument ที่ใช้ ซึ่งเราจะสามารถแบ่งตามระบบสัญญาณดังต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.ระบบลม (Pneumatic instrument) เป็นระบบที่รับสัญญาณค่าวัดจากตัววัด หรือตัวส่งผ่าน และส่งสัญญาณควบคุมไปควบคุม process ด้วยสัญญาณ pneumatic ที่ใช้เป็นมาตราฐานในปัจจุบันคือ 0.2-1.0 Kg/cm^2 (3-15 PSI) โดยใช้แรงลมป้อนที่ความดัน 1.4 Kg/cm^2 (20 PSI)

2.ระบบไฟฟ้า (Electronic instrument) เป็นระบบ instrument ที่รับสัญญาณค่าวัดจากตัววัดหรือตัวส่งผ่านและส่งสัญญาณไปควบคุม process ด้วยสัญญาณทางไฟฟ้า สัญญาณไฟฟ้าที่ใช้ในปัจจุบันเป็นมาตราฐานใหม่ สัญญาณไฟฟ้ามาตราฐานนี้คือ 4-20 mA

3.Hydraulic Instrumentation เป็นระบบที่รับสัญญาณจากตัวรับและส่งสัญญาณไปควบคุม process ด้วยสัญญาณ hydraulic ปัจจุบันมีใช้ค่อนข้างน้อย และมักเป็น instrument แบบควบคุมที่จุดวัด และใช้เพื่อจุดประสงค์บางประการเท่านั้น ระบบที่นิยมใช้มากที่สุดคือระบบไฟฟ้า

เราได้ทราบถึงลักษณะที่ใช้ในการควบคุมแบบที่การป้อนกลับแล้ว และยังได้กล่าวถึงค่าตัวแปรหรือสัญญาณต่างๆ ที่ถูกนำมาใช้หรือมีผลในการควบคุม เราจะนำพื้นฐานเหล่านี้เพื่อศึกษาตัวควบคุมในบทต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6

PID CONTROLLER

ในระบบควบคุมทางด้านอุตสาหกรรมส่วนใหญ่ ในระบบควมคุมย่อยจะเป็นแบบง่ายๆ ที่ใช้หลักการควบคุมแบบป้อนกลับ และโดยทั่วไปตัวควบคุมที่ใช้มักจะเป็นแบบ PID นี่เอง เราจึงต้องศึกษาให้เข้าใจถึงการควบคุมชนิดนี้ก่อนที่จะนำ PID Controller ไปใช้งาน หรือสร้างมันขึ้นมา เราจะกล่าวถึงการควบคุมชนิด P, PID, PD, PI ตามลำดับ

6.1.Proportional Control

คือการควบคุมที่กำหนดให้สัญญาณควบคุมมีขนาดเป็นสัดส่วน เหมาะสมกับขนาดของความผิดพลาด กล่าวคือ มันจะรับเอาค่าตัวแปรโปรเซสมาเปรียบเทียบกับค่าเป้าหมาย (SP) ได้ค่าความผิดพลาดออกมา ถ้าหากค่าความผิดพลาด (หรือบางทีอาจเรียกว่า ค่าเบี่ยงเบน) มีค่ามาก สัญญาณควบคุมหรือ Manipulated ก็จะมาก และในทางกลับกันถ้าค่าความผิดพลาดมีค่าน้อย Manipulated ก็จะมีค่าน้อยเป็นสัดส่วนตามกันไป ความสัมพันธ์ระหว่างสัญญาณขาออก M(t) กับสัญญาณขาเข้า (PV) ที่มีค่า SP ใดๆ แสดงได้ดังนี้คือ

สมการ 6.1

$$M(t) = \left(\frac{100}{PB}\right) \times e(t) + b$$

เมื่อ M(t) = สัญญาณขาออก (Control signal)

e(t) = ค่า Error หรือ Diviation (ค่าเป้าหมาย - ค่าตัวแปรโปรเซส)

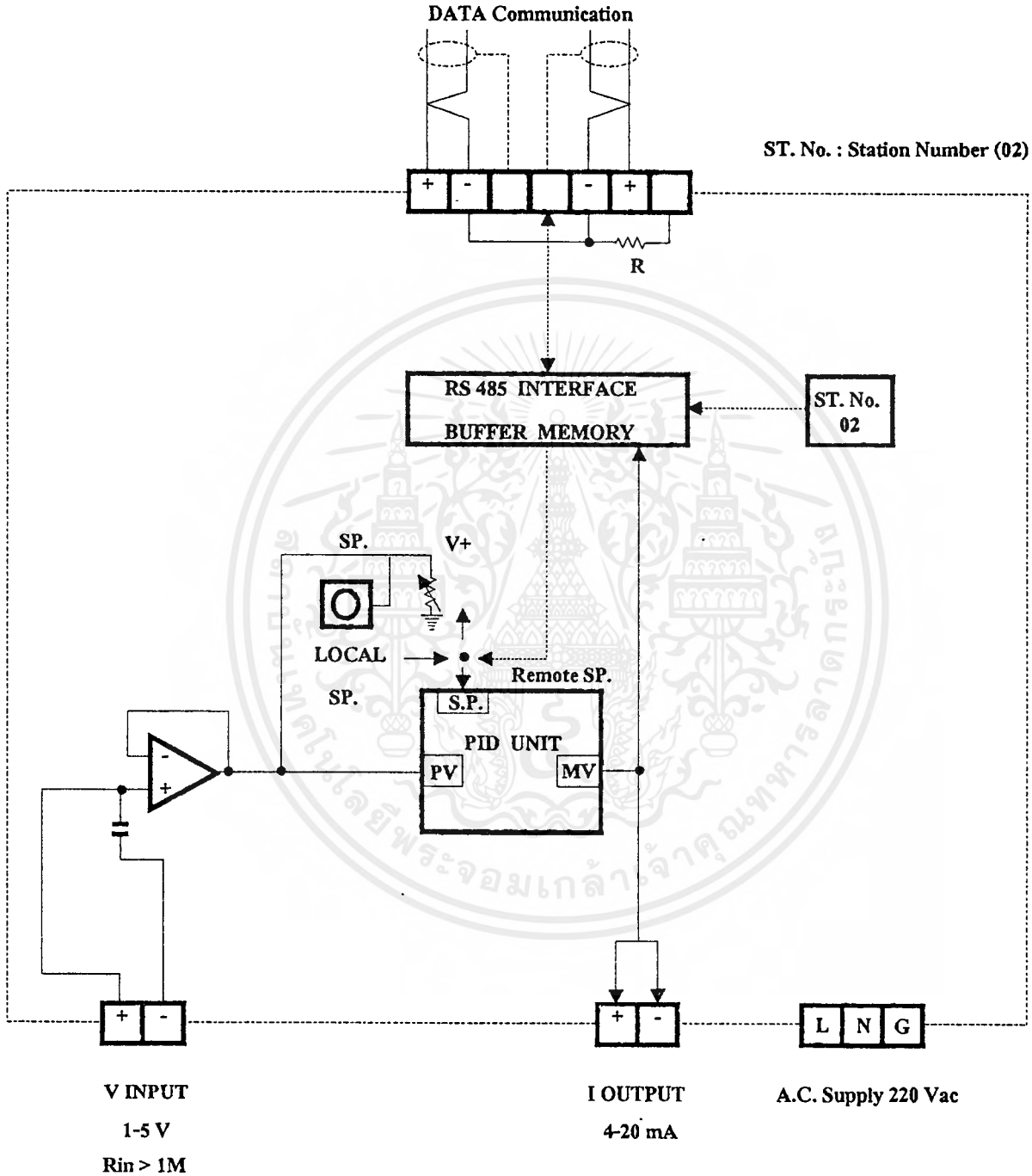
b = ค่าไบแอส (Bias)

PB = Proportional Band (หน่วยเป็น %)

คำจำกัดความของ Proportional band คือ เป็นเปอร์เซ็นการเปลี่ยนแปลงของสัญญาณขาเข้า (กรณีค่าเป้าหมายคงที่) ที่ทำให้สัญญาณขาออกเปลี่ยนแปลงไป 100% ผลการเปลี่ยนแปลงค่า PB ในกรณีที่ค่าเป้าหมายและค่าไบแอสมีค่า = 50% มีดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PID Control Unit

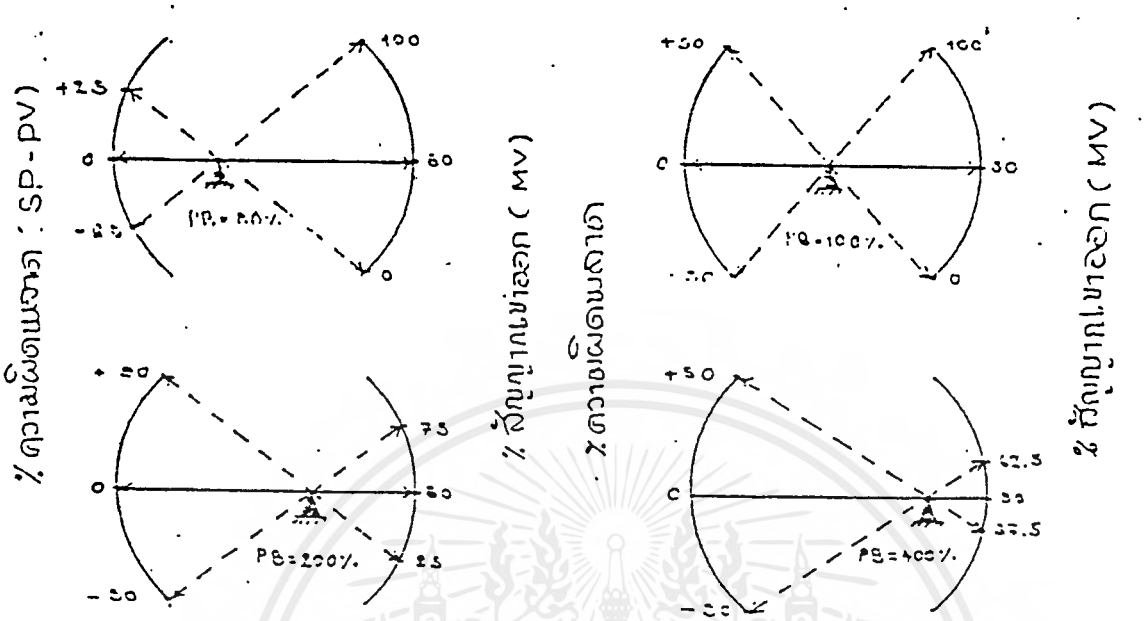


V INPUT
1-5 V
Rin > 1M

I OUTPUT
4-20 mA

A.C. Supply 220 Vac

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6.1 ผลของการปรับค่า PB

ข้อเสียของการควบคุมแบบ P อย่างเดียวคือ ถ้าสภาวะการทำงานและสภาพแวดล้อมเปลี่ยนแปลงไปจากค่านี้ (เกิด Disturbance) สัญญาณวัดจะมีค่าต่างไปจากเป้าหมายที่สภาวะสมดุลย์ใหม่ เรียกว่าเกิด offset ขึ้นแก้ไขโดยใช้ PI control

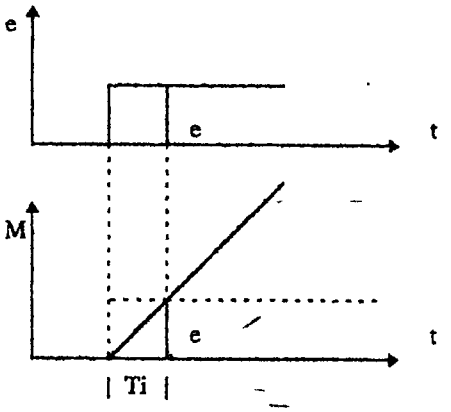
6.2.Integral Control (Reset control)

ความสัมพันธ์ระหว่างสัญญาณขาเข้าและสัญญาณขาออก จะมีความสัมพันธ์กันดังนี้

สมการ 6.2 $$M(t) = Ki * \int e(t)d(t)$$

แสดงว่าค่า Controller output เป็นปฏิภาคโดยตรงกับค่า Integral ของ Diviation นั่นเอง โดยที่ Ki ก็คือ อัตราขยายของการ Integral ซึ่งคือค่าคงที่ที่ใช้ในการปรับแต่ง (Adjustable Constant)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6.2 ผลของการ Integral time

จากรูปเราจะเห็นได้ว่าขณะที่ค่า e (deviation) มีการเปลี่ยนแปลงและไปคงที่อยู่ที่ค่าหนึ่ง ซึ่งเป็นลักษณะของการเกิด offset นั่นเอง ค่าของ manipulated ก็จะมีค่าสูงขึ้นไปเรื่อยๆ เพื่อเป็นการเร่งโปรเซสให้ปรับค่าจนค่า e เข้าใกล้ศูนย์เร็วขึ้น เรียกอีกอย่างว่าการ Reset เราจะสามารถพิสูจน์ได้ว่าค่า Ki = 1/Ti ดังนั้นเราจะเขียนสมการ 6.2 ได้ใหม่คือ

สมการ 6.3

$$M(t) = \left(\frac{1}{Ti}\right) \times \int e(t)dt$$

เมื่อ Ti คือ Integral time มีหน่วยเป็นเวลา อย่างไรก็ตามการควบคุมแบบ Integral นี้ก็มีข้อควรระวังในการกำหนดค่า Ti ว่าต้องมีความเหมาะสมกับโปรเซสที่ใช้ เพราะเมื่อค่า Ti น้อย Reset action ก็จะเกิดเร็ว (จากรูปที่ 6.2 ค่า m จะสูงขึ้นอย่างรวดเร็ว) จึงทำให้สัญญาณเบี่ยงเบนหมดเร็ว แต่ถ้าหากค่า Ti มีค่าน้อยไปจะเป็นเหตุให้ Damping ratio ของระบบมีค่าต่ำลง ซึ่งเป็นเหตุให้ระบบขาดเสถียรภาพ (Unstable) ได้ และถาค่า Ti มากเกินไป กว่าที่ระบบจะเข้าสู่เสถียรภาพก็จะใช้เวลานาน

6.3.Proportional plus Integral Control (PI Control)

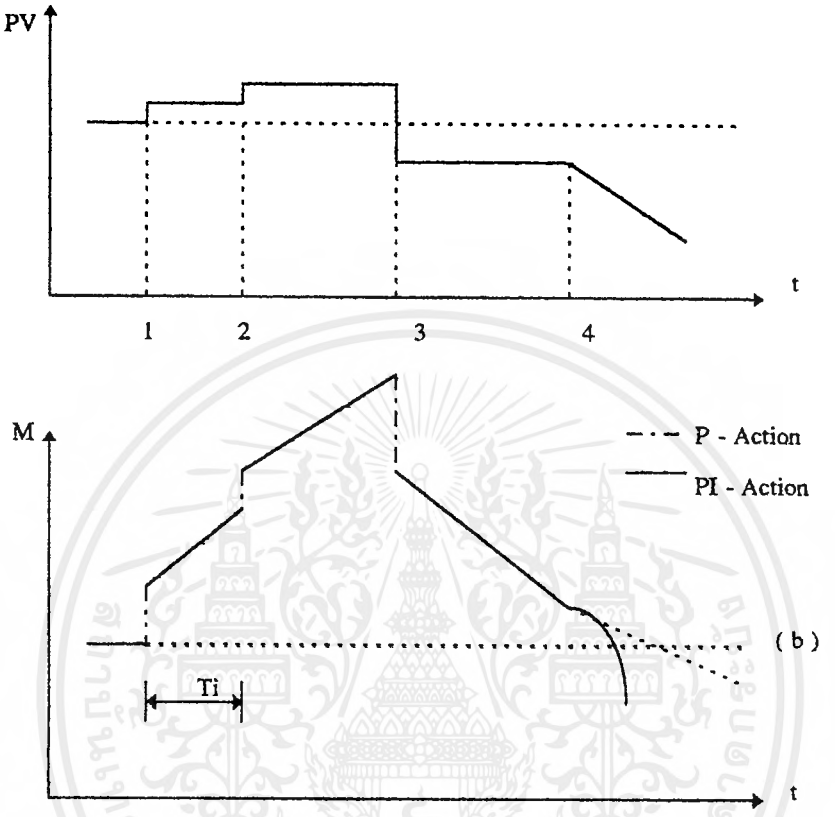
เราจะนำข้อดี P-Control และ I-Control มาใช้ร่วมกันเพื่อให้เกิดการควบคุมชนิดสัดส่วนของ P และสามารถกำจัดผลของ offset ได้ด้วยการควบคุมแบบ Reset ของ I-Control รวมกันแล้วได้สมการดังนี้คือ

สมการ 6.4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$M(t) = \left(\frac{100}{PB}\right)\left[e(t) + \frac{1}{Ti}\int e(t)dt\right] + b$$

พิจารณาลักษณะการตอบสนองของ MV และ PV แสดงได้ดังรูป 6.3



รูป 6.3 PI - Control

ขนาดของ Integral control ขณะใดขณะหนึ่งจะไม่สัมพันธ์กับค่าผิดพลาดขณะนั้น แต่จะเป็นสัดส่วนโดยตรงกับขนาดของความผิดพลาดสะสม $\left[\int (sp - pv)dt\right]$ ผลของ Integral action จะเหมือนกับการปรับค่าไบแอสจนกระทั่งค่าความผิดพลาดหมดไป ดังที่กล่าวในหัวข้อ I-Action มาแล้วว่า เมื่อ Ti มีค่าน้อยเกินไป ระบบอาจจะขาดเสถียรภาพอันเนื่องมาจาก Reset Action ที่เพิ่มเข้ามาส่งผลให้เกิดอาการ Cycling และถ้าหากว่าระบบของเรามีค่า Time lag ของเรามีค่ามากๆ แล้ว วิธีนี้ก็จะไม่เหมาะสมเพราะ การตอบสนองของตัวแปรของระบบจะช้ามากการสั่นกระเพื่อม (Hunting) ของค่าควบคุมจะทอดยาวยิ่งขึ้นมากกว่าเดิมไปอีก ไม่เข้าสู่ค่าเป้าหมายเร็วเท่าที่ควร อาจจะไม่สามารถแก้ไขข้อผิดพลาดได้ทันเวลา ดังนั้นระบบการควบคุมแบบ PI-Control จึงเหมาะสมกับโปรเซสที่ค่อนข้างไวอยู่แล้ว คือ Deadtime และ Capacity lag มีขนาดเล็ก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.4.Derivative Control

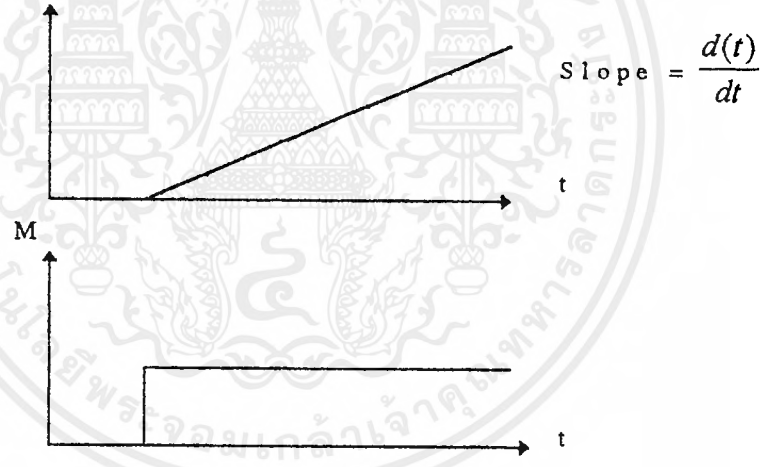
ผลของ Derivative Control ก็คือ มันจะให้สัญญาณควบคุมการตอบสนองแปรตามอัตราการเปลี่ยนแปลงของสัญญาณเบี่ยงเบน ดังสมการ 6.5

สมการ 6.5

$$M(t) = Td\left(\frac{de(t)}{dt}\right)$$

เมื่อ Td = derivative time

พิจารณาการตอบสนองระหว่าง deviation e และ Mv ดังรูป 6.4



รูป 6.4 Derivative Control

การควบคุมแบบนี้ เหมาะกับ Process ที่มี time lag มากๆ เพราะสามารถแก้ไขข้อผิดพลาดโดยการกระทำล่วงหน้าก่อนจะมีการผิดพลาดเกิดขึ้นจึงมีผลในการหยุดการเปลี่ยนแปลงของตัว process การกระทำล่วงหน้านี้อาศัยตัวแปรหนึ่งคือ derivative time หรือเวลาที่สัญญาณควบคุมนำหน้า สัญญาณเบี่ยงเบนนั่นเอง

รูปที่ 6.4 ค่าสัญญาณเบี่ยงเบนที่เป็น ramp signal เป็นสัญญาณเบี่ยงเบนที่เกิดจาก process ที่มี time lag มากๆ D-action จะมีผลก็ต่อเมื่อสัญญาณเบี่ยงเบนมีการเปลี่ยนแปลง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แบบเชิงเส้น ถ้าสัญญาณเบี่ยงเบนคงที่ D-action จะไม่มีผลทันที ด้วยเหตุผลนี้จึงต้องใช้ derivative control action ร่วมกับ proportional control หรือทั้ง proportional integral control และ derivative control จะได้ผลดียิ่งขึ้น ถ้าใช้ Td มากๆ เพราะเป็นการแก้ไขความผิดพลาดล่วงหน้า

ข้อเสียของ D-action คือ มีความไวต่ออัตราการเปลี่ยนแปลงสัญญาณเบี่ยงเบนมาก โดยเฉพาะ Td มาก อาจทำให้ระบบขาดเสถียรภาพจึงไม่เหมาะสมกับ process ที่มี time lag น้อย และตัวแปรของระบบเปลี่ยนได้ง่าย เช่น การควบคุมของไหล , ความดัน เป็นต้น

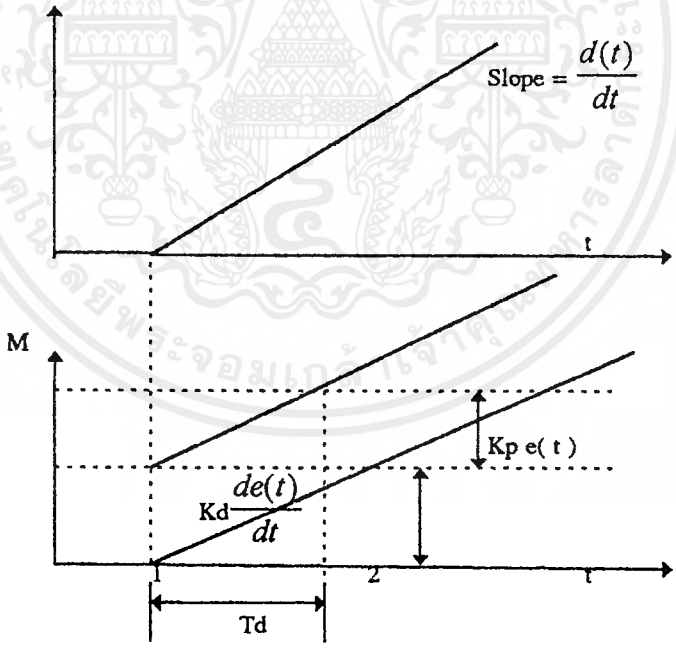
6.5.Proportional - plus - Derivative - Control (PD - Control)

ลักษณะความสัมพันธ์ของสัญญาณขาออก กับสัญญาณขาเข้าของตัวควบคุม PD - Controller เป็นไปตามสมการ 6.6 ดังนี้

สมการ 6.6

$$M(t) = \frac{100}{PB}\left[e(t) + Td\left(\frac{de(t)}{dt}\right)\right] + b$$

ถ้าสมมุติว่าที่เวลา t_1 Controller ได้รับค่าสัญญาณเบี่ยงเบนที่เป็น ramp function ที่มีอัตราการเพิ่มขึ้นเป็น de/dt (t) และ Control signal มีการแปรค่าไปดังรูปที่ 6.5



รูปที่ 6.5

ที่เวลา t_1 Derivative term จะทำให้ได้ kd de/dt(t) และที่เวลา t_2 proportional term จะทำให้ได้ term $K_p e(t)$ ซึ่งจะเพิ่มขึ้นจนเท่ากับ $K_d de/dt(t)$ พอดีซึ่งช่วงเวลา t_1-t_2 นี้เรียกว่า Derivative

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

time เนื่องจาก K_d de/dt(t) เพิ่มขึ้นอย่างเป็นเชิงเส้นจาก t_1-t_2 ด้วยสโลป de/dt(t) PD-Control นี้จะใช้ในการควบคุม process จำพวกเดียวกับ ที่ใช้ในการควบคุมชนิด P อย่างเดียว โดย PD-Control จะให้ค่า offset น้อยกว่า การควบคุมชนิด P อย่างเดียว

6.6.Proportional - Plus - Integral - Plus - Derivative Control (PID Control)

ในทางอุตสาหกรรมเรามักจะเรียกว่า "Proportional - Plus - Reset - Plus - Rate - Control" ดังที่กล่าวมาแล้วว่า Integral action ใน PI-Control จะทำให้ตัวควบคุมตอบสนองต่อความผิดพลาดช้าลง กรณีที่โปรเซสช้าอยู่แล้วเราอาจจำเป็นต้อง เพิ่ม Derivative action (หรือ Rate action) เพื่อลดความช้าของระบบควบคุม การควบคุมแบบนี้เราเรียกว่า PID-Control

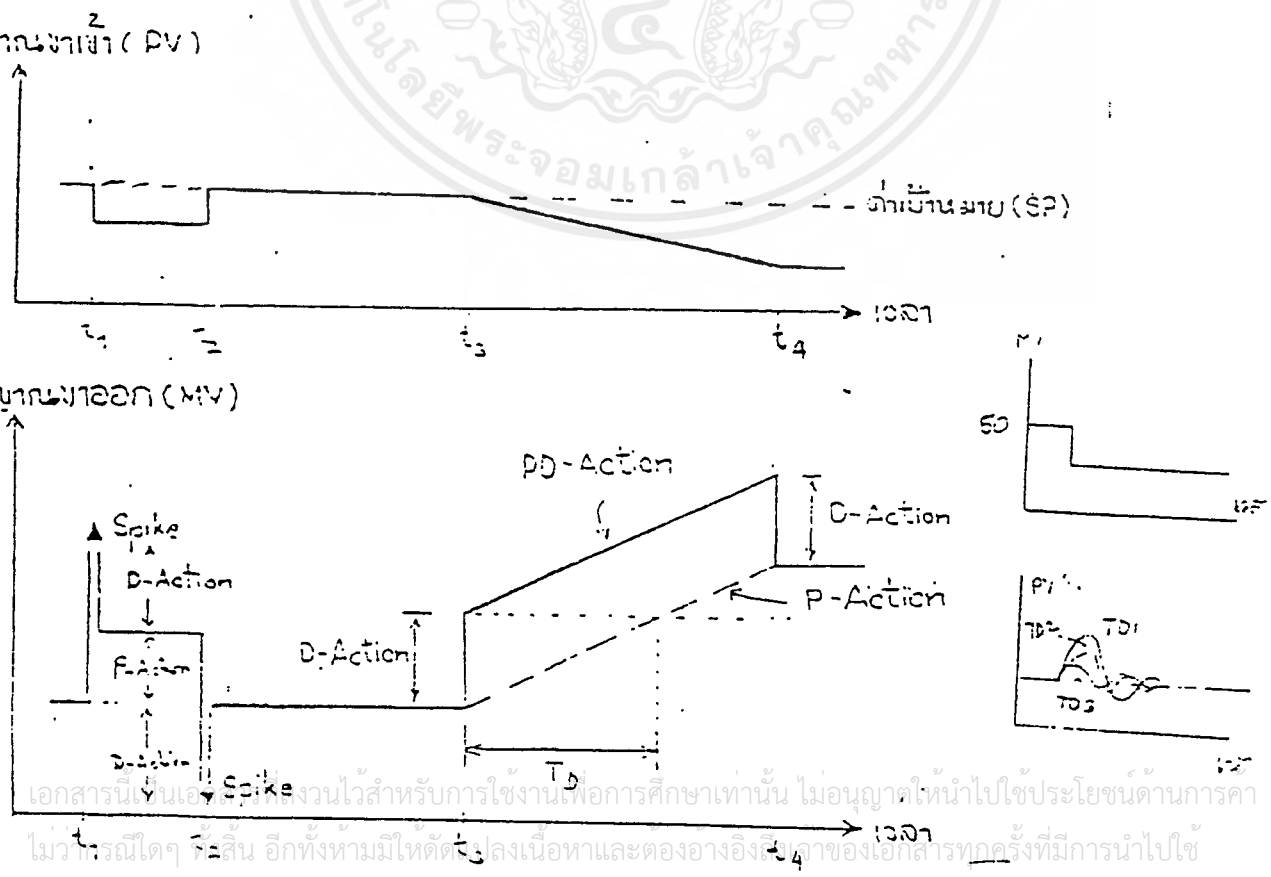
เราจะพบการควบคุมแบบ PID ในงานควบคุมอุณหภูมิเป็นส่วนใหญ่ ความสัมพันธ์ระหว่างสัญญาณขาออก (MV) กับสัญญาณขาเข้า (PV) ของ PID-Controller ที่ค่าเป้าหมาย (SP) ใดๆ แสดงได้ดังนี้

สมการ 6.7

$$M(t) = \frac{100}{PB}\left[e(t) + \left(\frac{1}{Ti}\right)\int e(t)dt + Td\left(\frac{de(t)}{dt}\right)\right] + b$$

Td = Derivative time หรือ Rate timeหน่วยเป็นนาที (Minutes)

กราฟแสดงความสัมพันธ์ของ MV กับ PV ในสมการ 6.7 จะยุ่งยากสับซ้อนไม่เหมาะกับการอธิบาย Derivative action รูปที่ 6.6 แสดงความสัมพันธ์ของ MV กับ PV ของ PD-Controller กรณี Open-loop ซึ่งง่ายต่อการทำความเข้าใจ Derivative action



เอกสารนี้เป็นเอกสารสงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 6.6 ผลของ D-Action ในตัวควบคุม PD Controller ขณะ Open loop

ผลตอบสนองของ Derivative action จะเป็นสัดส่วนโดยตรงกับอัตราการเปลี่ยนแปลงของค่า ความผิดพลาด (SP-PV)

กรณีที่ค่าเป้าหมายมีค่าคงที่ ตราบใดที่สัญญาณขาเข้าไม่เปลี่ยนแปลง Derivative action จะไม่มีผล ต่อสัญญาณขาเข้ามีการเปลี่ยนแปลง Derivative action จะเพิ่มหรือลดสัญญาณขาออกตามอัตราการเปลี่ยนแปลงของสัญญาณขาเข้า

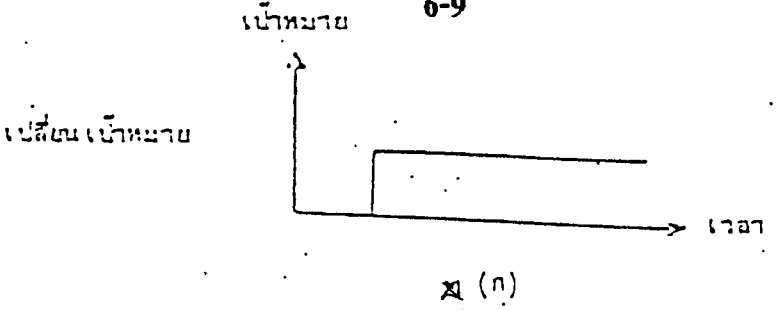
จากรูปที่ 6.6 ขณะเวลา ($t_3 < t < t_4$) สัญญาณขาเข้าเปลี่ยนแปลงด้วยอัตราที่คงที่ D-action จะทำงานคล้ายกับเป็นการทำให้ผลตอบสนองของ P-action เร็วกว่าเดิมเป็นเวลา Td นาที

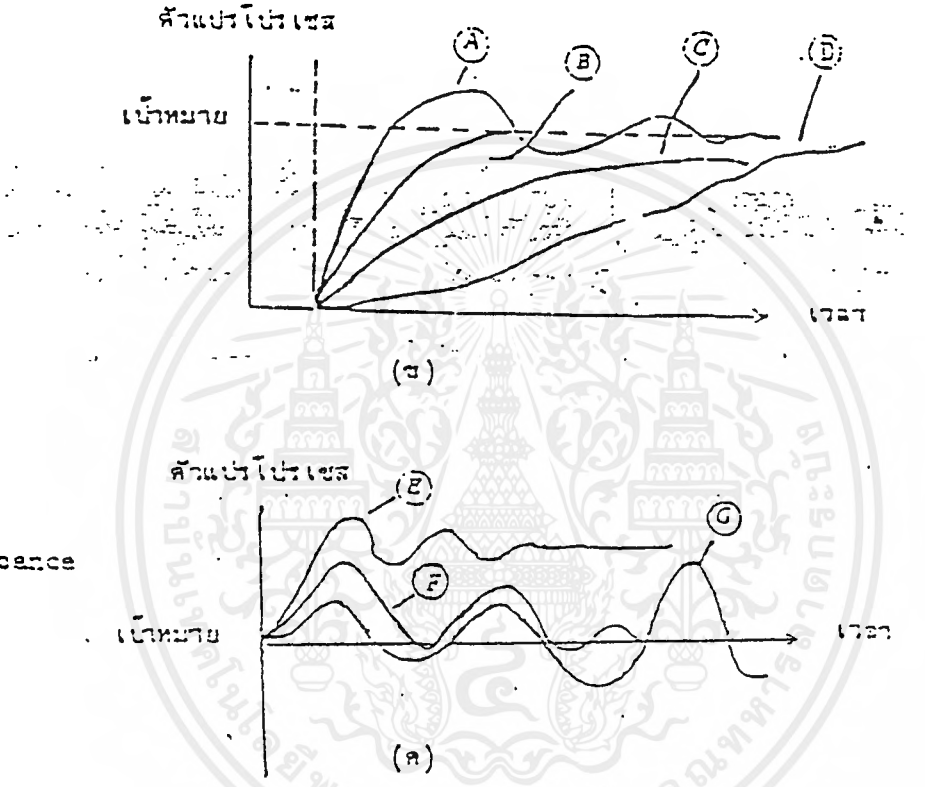
การเพิ่ม Derivative action เข้าไปใน PI-Control จะทำให้ผลตอบสนองของระบบต่อสิ่งรบกวนไวขึ้น สัญญาณวัดเปลี่ยนแปลงสู่ค่าเป้าหมายไวขึ้นจากรูป เราจะพบว่ายิ่ง Td มีค่ามาก สัญญาณวัดยิ่งเปลี่ยนแปลงกลับสู่ค่าเป้าหมายไวขึ้น อย่างไรก็ตาม ถ้าเราปรับตั้งค่า Td มากเกินไป สัญญาณวัดอาจเกิดการแกว่งได้ (กรณี $Td = Td_3$) ปกติเรามักปรับค่า Td ให้มีค่าใกล้เคียงกับค่า Td_2

6.7 การปรับค่า PID

จากการควบคุมทั้ง6ชนิดที่ได้กล่าวมาแล้วนั้น ต่างมีจุดหมายเดียวกัน คือ พยายามรักษาให้ค่าตัวแปรโปรเซสมีค่าเท่ากับค่าเป้าหมายอยู่เสมอ ในกรณีที่เกิด disturbance ในระบบ หรือมีการเปลี่ยนค่าเป้าหมายใหม่ จะทำให้ตัวแปรโปรเซสมีค่าต่างจากค่าเป้าหมายในขณะหนึ่ง ตัวควบคุมก็จะพยายามควบคุมให้ตัวแปรโปรเซส มีค่าต่างจากค่าเป้าหมายในขณะหนึ่ง ตัวควบคุมก็จะพยายามควบคุมให้ตัวแปรโปรเซสพยายามเข้าใกล้ค่าเป้าหมายนี้ในที่สุด ซึ่งลักษระการนำค่าตัวแปรโปรเซสเข้าใกล้ค่าเป้าหมายนี้ จะแตกต่างกันตามคุณสมบัติของระบบควบคุม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้





รูปที่ 6.7 ผลตอบสนองของระบบควบคุมต่างๆ

ระบบควบคุมในรูปที่ 6.1 ถ้าลองเปลี่ยนค่าเป้าหมาย ตัวควบคุมจะพยายามควบคุมให้ตัวแปรโปรเซสวิ่งเข้าหาค่าเป้าหมายนั้น ผลตอบของตัวแปรโปรเซสจะมีหลายแบบดังแสดงในรูป 6.7 ข.ดังนี้

A มี Overshoot และ การแกว่ง

B ตอบรับเร็ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

C ตัวแปรโปรเซสมีค่าไม่เท่ากับค่าเป้าหมายแม้เวลาจะผ่านพ้นไปนาน เรียกว่าเกิด offset

D ตอบรับช้ามาก

เมื่อตัวแปรโปรเซสหยุดนิ่งที่ค่าเป้าหมาย ในขณะนั้นเกิดมี disturbance เข้ามารบกวนโปรเซส (disturbance ได้แก่ การเปลี่ยนของโหลดการเปลี่ยนแปลงของสภาพแวดล้อม เป็นต้น) ผลตอบสนองของโปรเซส จะมีหลายแบบ ดังแสดงในรูป 6.7 ค

E เกิด offset ได้ค่าผิดพลาดไปจากค่าเป้าหมายเดิม

F เกิดการแกว่งเล็กน้อย ก่อนการกลับสู่ค่าเป้าหมายเดิม

G เกิดการแกว่งและขาดเสถียรภาพ

เมื่อพิจารณาผลตอบสนองของการควบคุมชนิดต่างๆ เหล่านี้จึงพอสรุปได้ว่า ระบบการควบคุมที่ดี จะต้องมีคุณสมบัติที่ดีดังนี้

ก) มีเสถียรภาพ ไม่เกิดการแกว่ง (Oscillation) เมื่อถูกกระตุ้น

ข) ตอบรับการเปลี่ยนค่าเป้าหมาย หรือ disturbance ได้รวดเร็ว

ค) ไม่เกิด offset

6.8 ความยากง่ายในการควบคุมของโปรเซส

ลักษณะทั่วไปของโปรเซสนั้นจะมี Time lag จากรูปคลื่นผลตอบสนองของโปรเซสต่อ Step input เราสามารถหาค่า dead time (LE) และค่าคงตัวเวลา (TE) โดยประมาณได้

อัตราส่วน LE/TE นี้จะเป็นค่าที่ใช้ประเมินความยากง่ายในการควบคุมและใช้เลือกแบบการควบคุม พิจารณาตาราง

LE/TE	แบบการควบคุมที่เหมาะสมกับโปรเซสนั้น
LE/TE < 0.2	ON/OFF ,P ,PI
0.2 < LE/TE <1.0	PI ,PID
1.0 < LE/TE	Feedforward ,Computer control

สาเหตุของการขาดเสถียรภาพในระบบ

ก)ผลของระบบควบคุมข้างเคียว หรือ อื่นๆ

ข)Disturbance เป็น periodic

ค)มี Nonlinear element ในระบบเองเช่น ความผิด dead band เป็นต้น

ง)Process gain หรือ Loop gain มีค่าสูงเกินไป

จ)คุณสมบัติของตัวแปรในระบบไม่คงที่แปรเปลี่ยนตาม Disturbance, ค่าเป้าหมายม , เวลา เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.9 ผลของ PID ต่อเสถียรภาพของระบบ

ในระบบควบคุมแบบป้อนกลับ ซึ่งใช้ตัวควบคุมแบบ PID นั้น ถ้าเราลองแปรค่า PB , Ti , Td จะมีผลต่อผลตอบสนองของระบบควบคุมดังนี้

1.ผลของ P action

เมื่อลดค่า PB ลง ทำให้อัตราขยายสูงขึ้น จะมีผลทำให้

ก) offset ลดลง

ข)period ของการแกว่งเล็กลง

ค)อัตราส่วนของช่วงกว้างการแกว่งเพิ่มขึ้น ระบบขาดเสถียรภาพมากขึ้น

2.ผลของ I action

เมื่อ P และ D action คงที่แล้ว ลองลด Ti (Reset time) จะมีผลทำให้

ก)offset จะหายไป

ข)ผลตอบสนองจะเร็วขึ้น (Fast response)

ค)อัตราส่วนของช่วงกว้างการแกว่งเพิ่มขึ้น ระบบขาดเสถียรภาพมากขึ้น

3.ผลของ D action

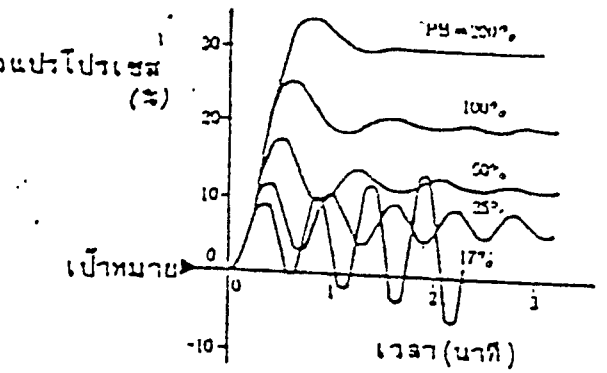
เมื่อให้ P และ I action คงที่แล้ว ลองเพิ่มค่าเวลา Td (Derivative time) ให้ยาวขึ้น จะมีผลทำให้

ก)อัตราส่วนของช่วงกว้างการแกว่งลดลง ระบบมีเสถียรภาพมากขึ้น

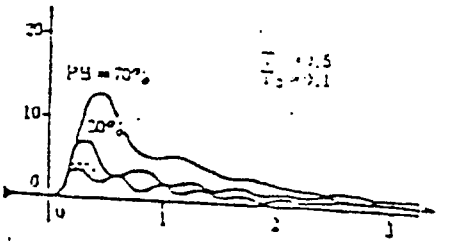
ข)period ของการแกว่งสั้นลง

ตามปกติการใช้ค่าเวลา Td ให้ยาว มีแนวโน้มที่จะทำให้ระบบมีเสถียรภาพมากก็จริง แต่ก็มีจุดอ่อนตรงตอบรับต่อ Noise ได้ง่าย ทำให้ผลตอบสนองของระบบไวเกินไป ผลตอบสนองของการควบคุมแบบต่างๆ ต่อการเปลี่ยนแปลงค่าเป้าหมายและะการเกิด Disturbance โดยตั้งค่า PB , Ti , และ Td ต่างกัน แสดงในรูป 6.8 และ 6.9

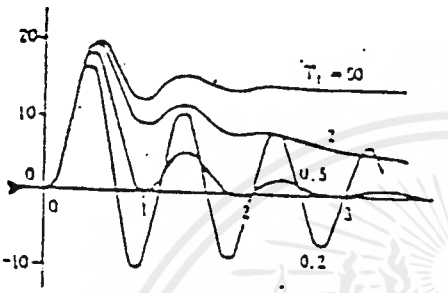
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



P Control เมื่อเปลี่ยน PB

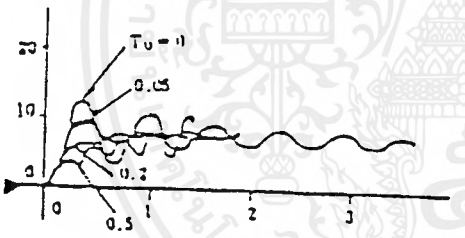


PID Control เมื่อเปลี่ยน PB

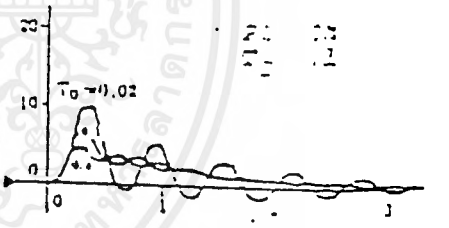


PI Control เมื่อเปลี่ยน T_I

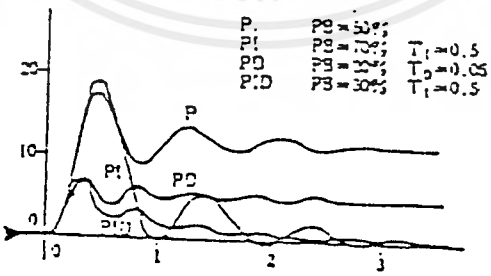
PID Control เมื่อเปลี่ยน T_I



PD Control เมื่อเปลี่ยน T_D



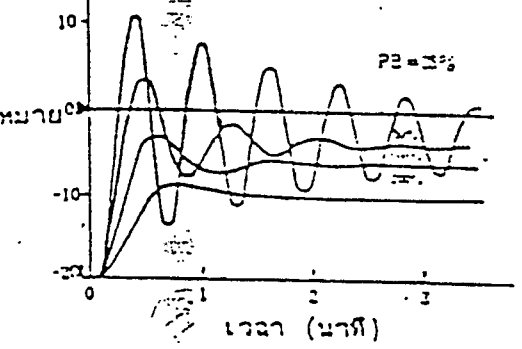
PID Control เมื่อเปลี่ยน T_D



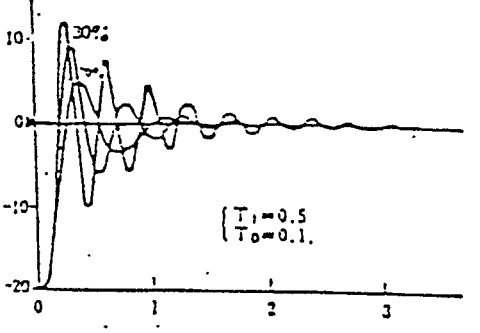
เปรียบเทียบผลตอบของแบบควบคุมต่างๆ

รูปที่ 6.8 ผลตอบสนองของระบบควบคุมแบบต่างๆ เมื่อมี Disturbance

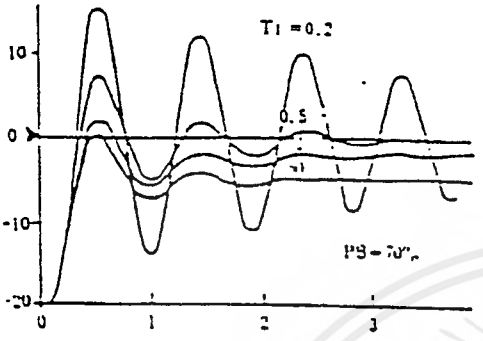
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



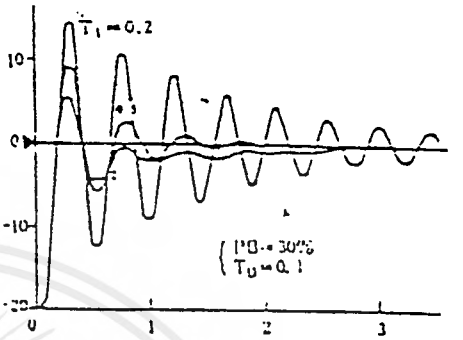
P. Control เมื่อเปลี่ยน PB



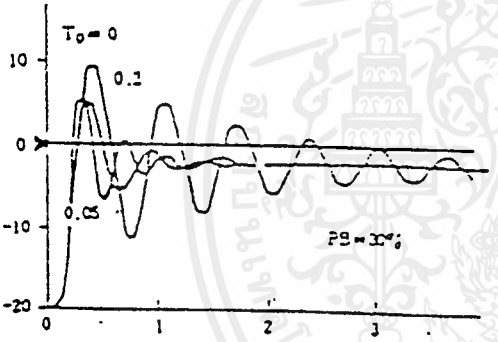
PID Control เมื่อเปลี่ยน PB



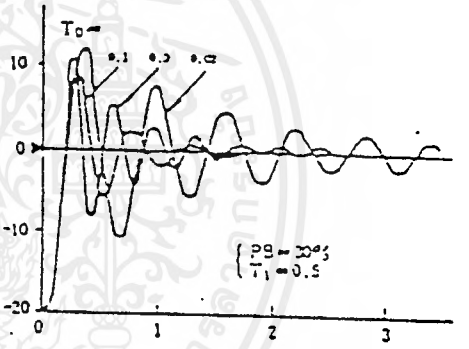
PI Control เมื่อเปลี่ยน T_I



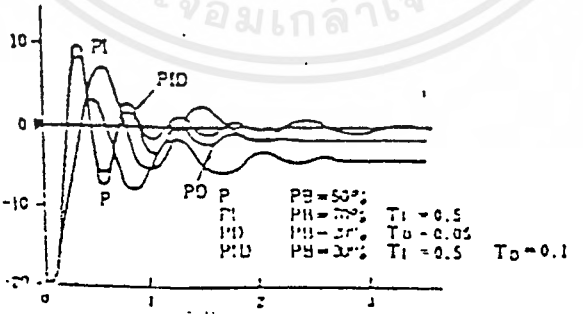
PID Control เมื่อเปลี่ยน T_I



PD Control เมื่อเปลี่ยน T_D



PID Control เมื่อเปลี่ยน T_D



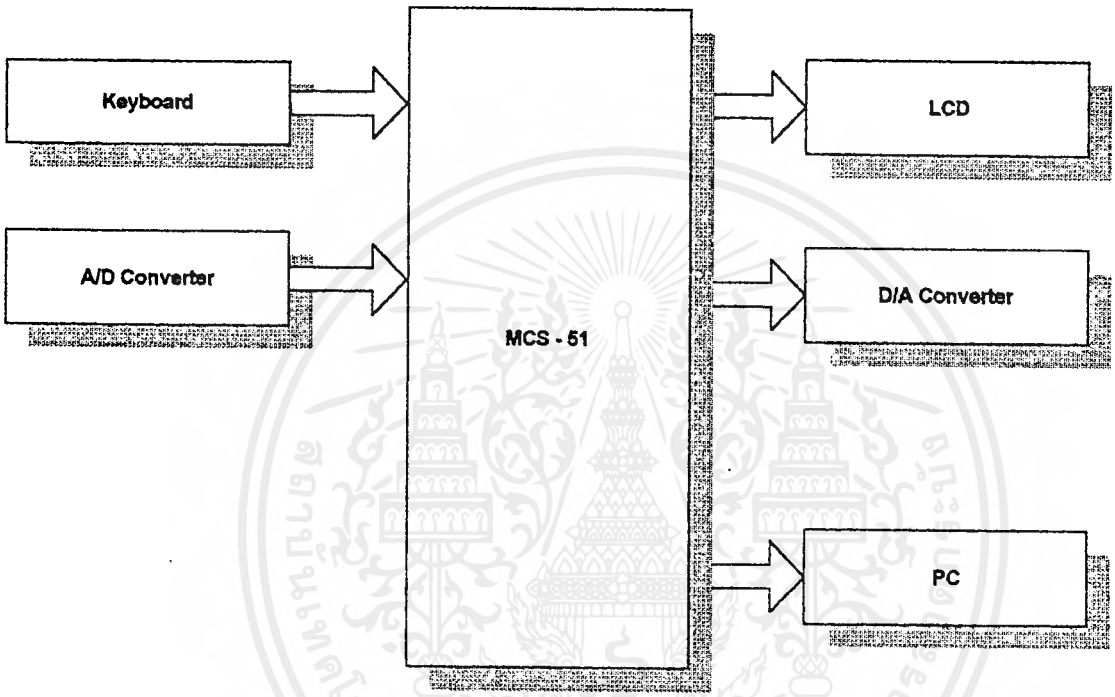
เปรียบเทียบผลตอบสนองของแบบควบคุมต่างๆ

รูปที่ 6.9 ผลตอบสนองของระบบควบคุมแบบต่างๆ เมื่อเปลี่ยนค่าเป้าหมาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนประกอบทางด้าน Hard Ware

ใน PID Controller นี้ มีส่วนประกอบทางด้าน Hard Ware ดังนี้คือ



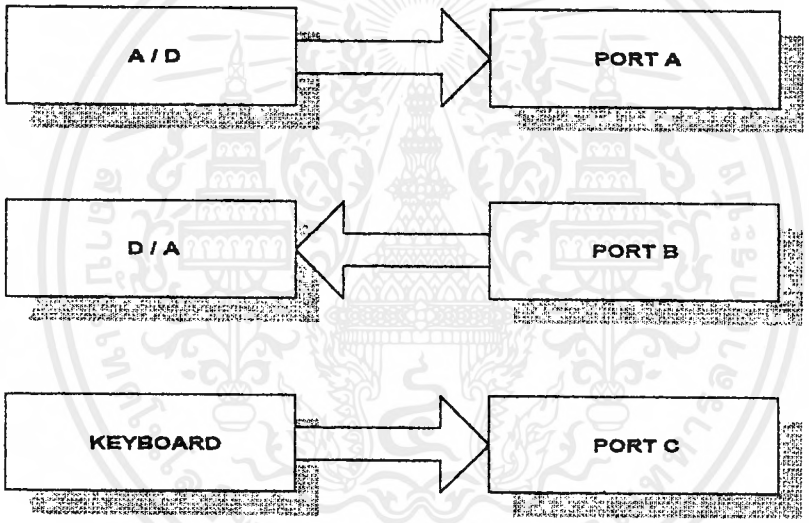
แผนผังส่วนประกอบทางด้าน Hard Ware

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.MCS-51

พิจารณาเลือกใช้ Board Controller รุ่น ANT-32 เบอร์ 80C32 ของบริษัทศิลารีเสิร์ช เนื่องจากมีองค์ประกอบที่ เหมาะสมสำหรับการใช้งานดังนี้ คือ

1.1 มีพอร์ทที่เป็น อินพุท/ เอาท์พุท อยู่ด้วยกัน 2 พอร์ท คือ 8255 ซึ่งสามารถที่จะกำหนดให้เป็นอินพุทหรือเอาท์พุทก็ได้ ในที่นี้เราพิจารณาเลือกใช้ User Port 1 โดยจะประกอบด้วย อินพุท/ เอาท์พุท 3 พอร์ท



โดยหน่วยความจำของ 8255 ที่ User Port 1 จะอยู่ที่

Address F800H + 8255 offset address = Actual address

Port A ตำแหน่งแอดเดรส F800H + 00H = F800H

Port B ตำแหน่งแอดเดรส F800H + 01H = F801H

Port C ตำแหน่งแอดเดรส F800H + 02H = F802H

Mode Port ตำแหน่งแอดเดรส F800H + 03H = F803H

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.2 มีหน่วยความจำที่เป็น Program and Data MeMory ขนาด 8-32 Kbyte ซึ่งหน่วยความจำที่ใช้ในการเขียนโปรแกรมการควบคุม PID นี้ มีขนาด 8 Kbyte จึงสามารถที่จะใช้ได้พอ

1.3 มีพอร์ทของ LCD ให้มาต่อได้เลย ทำให้มีความสะดวกในการใช้งาน โดยตำแหน่ง address ของ LCD จะอยู่ที่

Address	**ลักษณะของ Port ที่ติดต่อ**	
FA00H	สำหรับเขียนคำสั่ง	(RS = 0 R/W = 0)
FA01H	สำหรับอ่านค่า BUSY	(RS = 0 R/W = 1)
FA02H	สำหรับเขียนข้อมูล	(RS = 1 R/W = 0)
FA03H	สำหรับอ่านข้อมูล	(RS = 1 R/W = 1)

1.4 มีพอร์ทสื่อสารอนุกรม RS 232 ให้มาด้วย สำหรับการติดต่อกับ PC ทำให้สะดวกในการใช้งานติดต่อ

2. A/D Converter

ประกอบด้วยวงจรที่ทำหน้าที่ 2 ส่วน คือ

2.1 I/V Convertor

โดยจะรับสัญญาณกระแสมาตราฐาน 4 - 20 mA แปลงไปเป็นแรงดันไฟฟ้า 0 - 5 Vdc

2.2 A/D Converter

จากแรงดัน 0 -5 Vdc จะแปลงเป็นสัญญาณดิจตอล 00 - FFH โดยใช้ ไอซี แปลงจาก แรงดันไฟฟ้าเป็นดิจิตอล เบอร์ ADC 0801

3. D/A Converter

ประกอบด้วยวงจรที่ทำหน้าที่ 2 ส่วน คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1 D/A Converter

แปลงสัญญาณดิจิตอล 00 - FFH เป็นแรงดันไฟฟ้า 1 - 5 Vdc โดยใช้ไอซี เบอร์ DAC 0808 แปลง

3.2 V/I Converter

แปลงจากสัญญาณแรงดันไฟฟ้า 1 - 5 Vdc เป็นสัญญาณกระแสมาตราฐาน 4 - 20 mA เพื่อส่งไปควบคุมกระบวนการต่อไป

4.Keyboard

ใช้สวิทซ์แบบกดติดปล่อยดับ จำนวน 6 ตัว ต่อโดยตรงเข้ากับพอร์ท C ของ 8255 โดยจะรับค่า "1" เมื่อมีการกดสวิทซ์แต่ละตัว

5. LCD

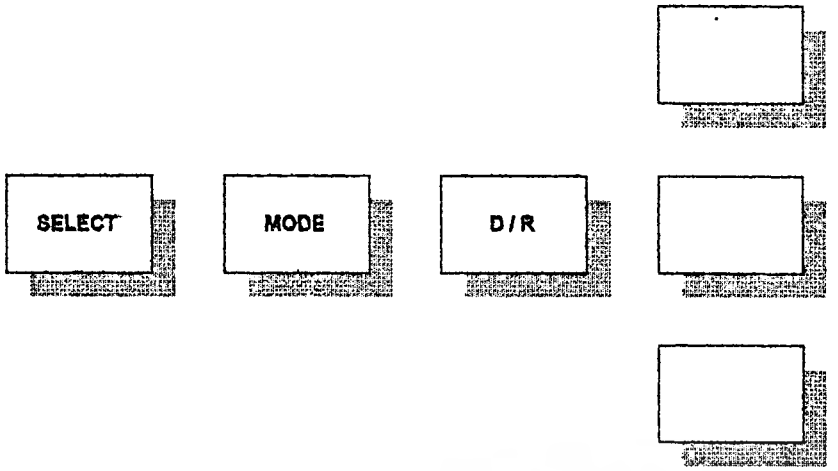
ใช้ LCD แบบ Dot Matrix เบอร์ DMC 202 มีตัวอักษร 20 ตัว และ มี 2 บรรทัด ต่อเข้ากับพอร์ท LCD ของ MCS-51 ได้เลย

6. PC

ส่งออก PC โดยผ่านทาง RS 232 แปลงเป็น RS 485 อีกที่หนึ่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

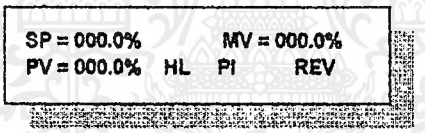
KEYBOARD



1. SELECT

ทำหน้าที่ในการเลือกเมนูในการ Set ค่าพารามิเตอร์ต่าง ๆ ซึ่งจะประกอบด้วยเมนู จำนวน 6 เมนู ดังนี้

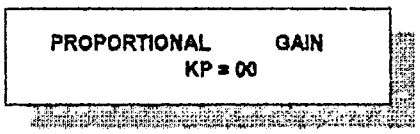
1.1 Control Menu



แป็นเมนูสำหรับใช้ในการควบคุม Process โดยจะแสดงค่าของสัญญาณ Set Point, Manipulate Variable และ Process Variable โดยจะแสดงค่าอยู่ในช่วงของ 0 -100.0 % รวมทั้งแสดงสภาวะของสัญญาณเตือน (Alarm) ได้แก่ High Alarm และ Low Alarm ซึ่งเป็นสัญญาณที่เปรียบเทียบกับค่าของ PV และ Alarm เมนูนี้สามารถแสดงสภาวะของการควบคุมได้อีก 6 โหมด อันประกอบด้วย การควบคุมในโหมด MAN, P, PI, PID, PD และ ON-OFF Control นอกจากนี้สามารถแสดง Action ของการควบคุมได้อีก คือ Direct และ Reverse Action

สำหรับการควบคุมต่าง ๆ นั้น จะใช้เมนูนี้ในการควบคุมกระบวนการ

1.2 เมนู Set ค่า KP

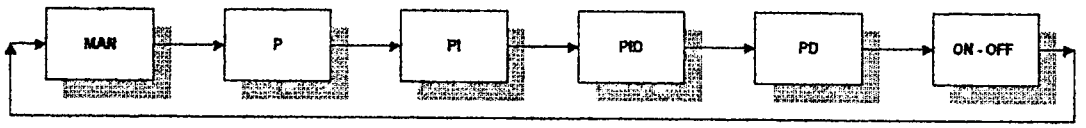


เมนูนี้สำหรับการ Set ค่า ของ Gain การควบคุม จะอยู่ในรูปของ Gain KP ซึ่งสามารถเลือกค่า KP ได้ ตั้งแต่ 0 - 9

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. MODE

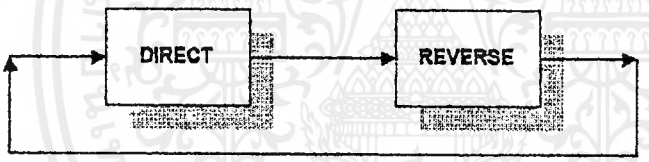
ทำหน้าที่ในการเลือกโหมดของการควบคุม โดยจะประกอบด้วย 6 โหมด คือ MAN, P, PI, PID, PD และ ON-OFF Control เมื่อทำการกดปุ่มนี้ โหมดการควบคุมจะเปลี่ยนไปเรื่อย ๆ และวนลูป กลับเหมือนเดิม



ลูปในการกดปุ่ม MODE

3. D/R

ทำหน้าที่ในการเลือก Action ของการควบคุมได้แก่ Direct และ Reverse โดยเมื่อกดปุ่มนี้แล้ว Action ของการควบคุมจะเปลี่ยนไปเรื่อย ๆ คือ Direct และ Reverse



ลูปในการกดปุ่ม D/R

4. ปุ่มเพิ่มและลดค่า

ทำหน้าที่ในการเพิ่มหรือลดค่าของ Set Point ถ้ากดปุ่ม จะเพิ่มอย่างช้า ด้วยอัตราเร็ว 0.2 ms และถ้ากดปุ่ม จะลดด้วยอัตราเร็ว 0.2 ms แต่ถ้ากดปุ่ม พร้อมด้วย การกดปุ่มหรือ จะทำการเพิ่มหรือลดอย่างรวดเร็ว ด้วยอัตราความเร็ว 0.02 ms

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การเขียนโปรแกรมควบคุม

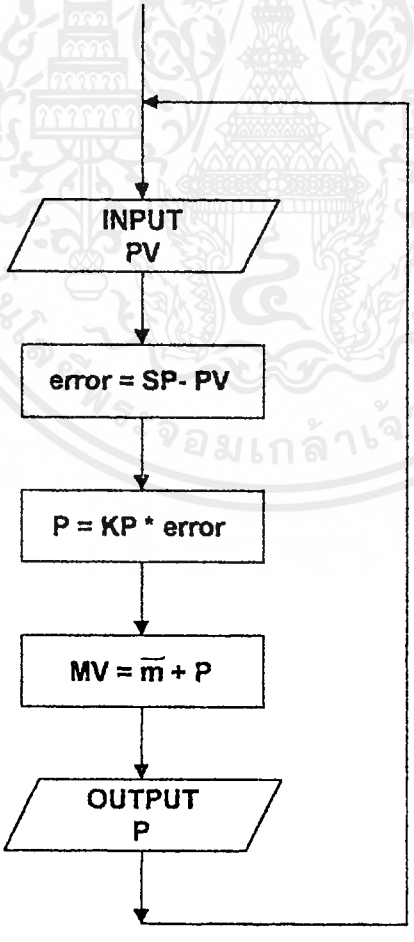
จากสมการของเทอม PID จะอยู่ในรูปของ

$$M(t) = m + Kp\ e(t) + Kp/Ti \int e(t).d(t) + Kp.Td.d\ e(t)$$

จากสมการนี้ จะพบว่า มีเทอมของคำสั่งทางคณิตศาสตร์อยู่คือ เทอมของ การบวก, การลบ, การคูณ และการหาร, การอินทิเกรท และ การดิฟเฟอรเรนเชียล

สำหรับคำสั่งของการบวก, การลบ, การคูณ และการหารนั้น สามารถที่จะทำได้ โดยใช้ภาษาแอสแซมบลีของตัว MCS-51 คือคำสั่ง ADD, SUBB, MUL และ DIV ที่ตัว MCS-51 รองรับมีอยู่แล้ว แต่สำหรับคำสั่ง อินทิเกรท และดิฟเฟอรเรนเชียล นั้น ไม่มีคำสั่ง นี้อยู่ใน MCS-51 ดังนั้น จึงต้องทำการแปลงคำสั่งเหล่านี้ให้อยู่ในคำสั่งการบวก, การลบ, การคูณ และ การหาร ที่ตัว MCS-51 เข้าใจ

1.การเขียนโปรแกรมในเทอม P



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Proportional Mode Flow Chart

จากสมการ OUTPUT P = m + Kp e(t)

จากสมการเทอม P สามารถแปลงให้อยู่ในรูปอัลกอริธึม ที่ตัว MCS-51 เข้าใจได้โดย

OUTPUT P = m + Kp(SP-PV) ——— !!!

เมื่อ OUTPUT P = สัญญาณเอาท์พุทของ Controller แบบ P %

m = ค่า Bias ของ Controller ที่ 50% ขณะไม่มีความผิดพลาด

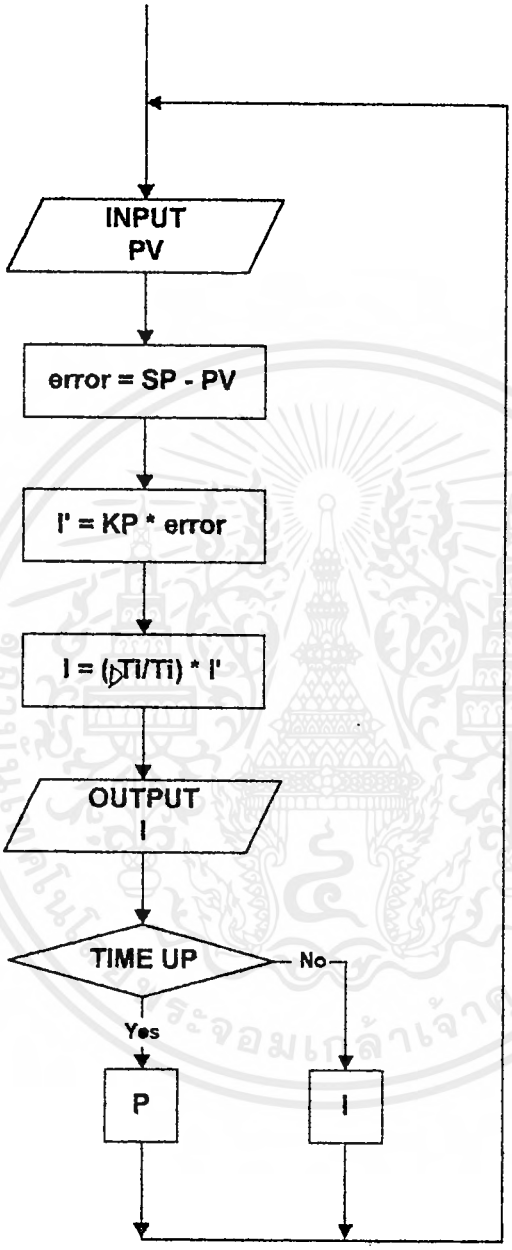
Kp = Proportional Gain

SP = Set Point Value %

PV = Process Variable %

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.การเขียนโปรแกรมในเทอม I



Integral Mode Flor Chart

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากสมการ OUTPUT I = $Kp/Ti \int e(t).d(t)$

แปลงให้อยู่ในรูปอัลกอริธึม ได้ คือ

$$OUTPUT\ I = (Kp/Ti).\Delta Ti\ (SP-PV) \text{ ——— !!!}$$

เมื่อ OUTPUT I = สัญญาณ OUTPUT ของ Controller I %

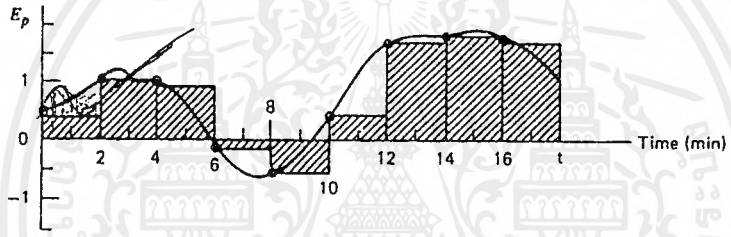
Kp = Proportional Gain

SP = Setpoint Value %

PV = Process Valiable %

ΔTi = Time Between Sample msec

Ti = Integral Time msec

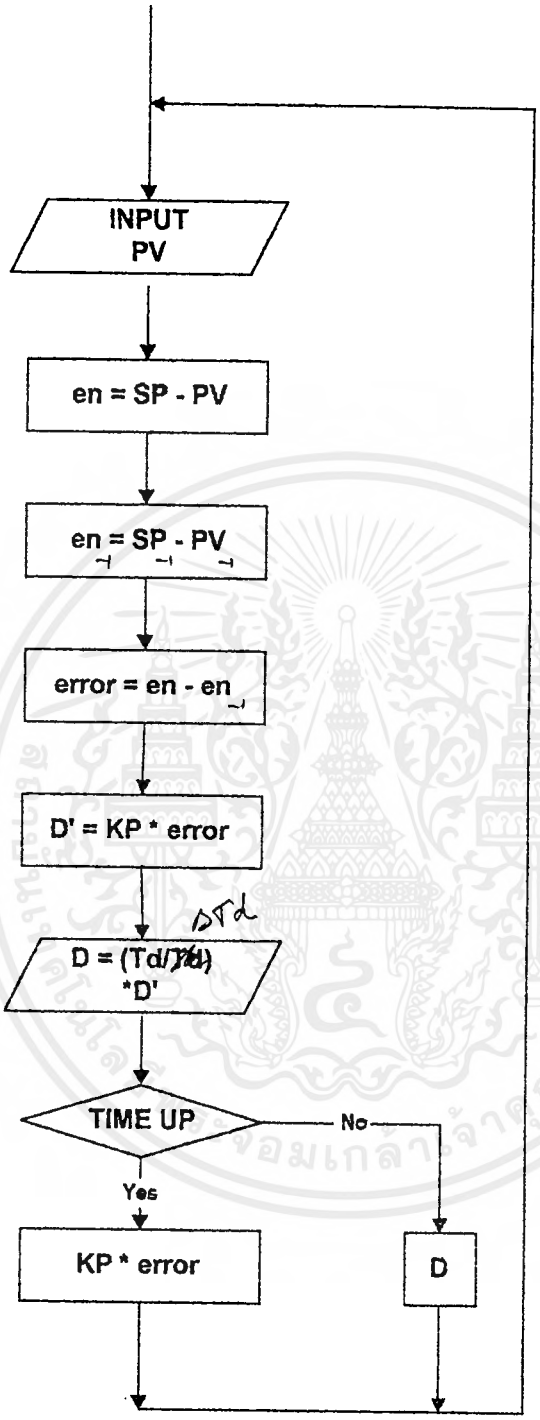


จากสมการและจากรูป จะพบว่า เทอม I คือการรวมค่าความผิดพลาดที่เกิดขึ้นทั้งหมดในช่วงเวลา Tiขณะที่ค่าของ ΔTi ยังมีค่าไมครบเวลาTi Controllerจะส่งสัญญาณเทอม I ที่เกิดจากค่าของI=Kp.(ΔTi/Ti) (SP-PV) ออกไปรวมกับเทอม P และเมื่อค่าของ ΔTi ครบเวลาของ Ti จะพบว่า Controller จะส่งค่าของสัญญาณเทอม I ออกไป มีค่าเท่ากับเทอม P ออกไปรวมกับเทอม P ออกไปควบคุม

$$I = Kp.(\Delta Ti/Ti)\ (SP-PV) = P$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.การเขียนโปรแกรมในเทอม D



Differential Mode Flow Chart

จากสมการ OUTPUT D = Kp.Td.d e(t)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แปลงให้อยู่ในรูป อัลกอริธึมได้

$$OUTPUT\ D = Kp.(Td/\Delta Td)\ (e_n - e_{n-1}) \text{ ——— !!!}$$

เมื่อ OUTPUT D = สัญญาณ OUTPUT ของ Controller แบบ D

Kp = Proportional Gain

$e_n = SP\text{-}PV$

$e_{n-1} = SP_{-1} - PV_{-1}$

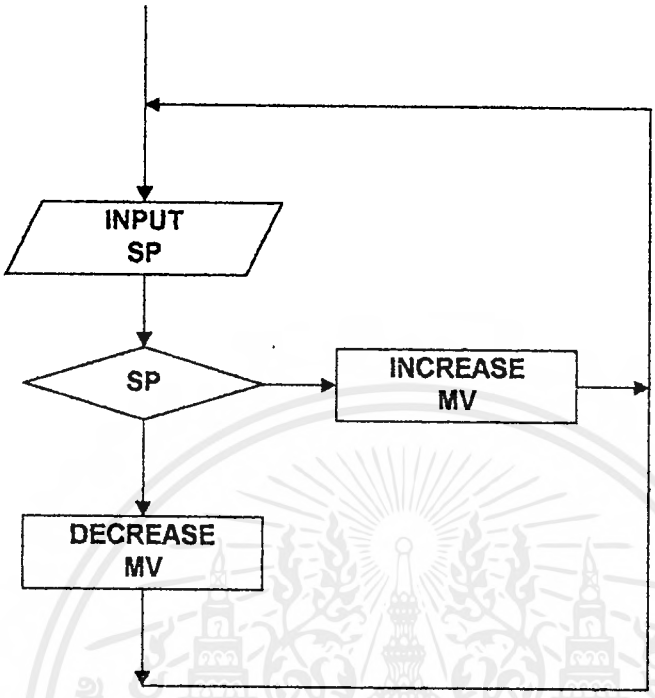
ΔTd = Time Between Sample msec

Td = Differential Time msec

เนื่องจากเทอม D คือการคาดคะเนล่วงหน้าของค่าความผิดพลาดที่เกิดขึ้น ดังนั้น จึงดูที่ความชัน (Slope) ของการเปลี่ยนแปลงความผิดพลาด ณ. ครั้งนี้ e_n เปรียบเทียบกับ ความผิดพลาด ณ. ครั้งที่ผ่านมา e_{n-1} นำมาเปรียบเทียบกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.การเขียนโปรแกรมในเทอม MAN

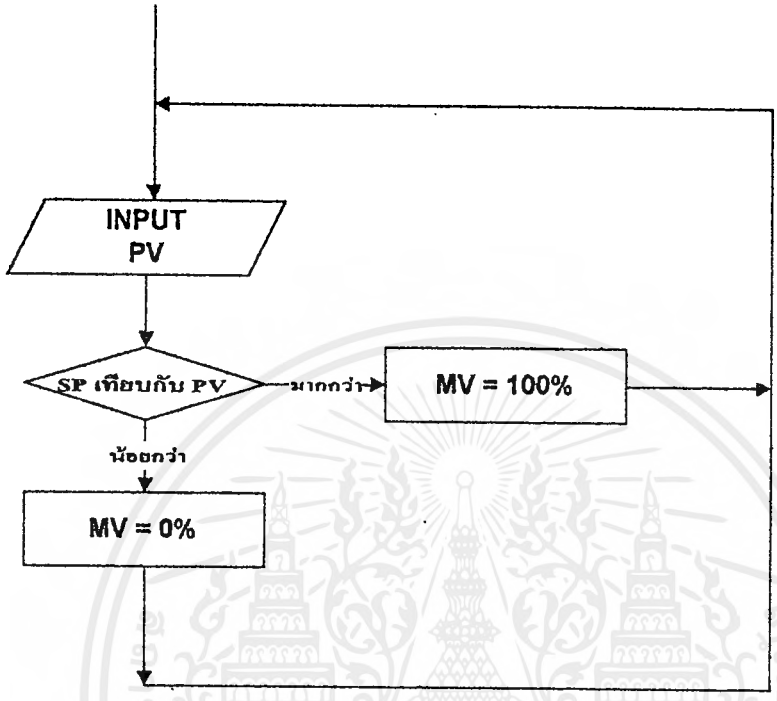


Manual Mode Flow Chart

การปรับค่า Set Point ฟนโหมด MAN จะเปลี่ยนค่า MV ตามการเปลี่ยนแปลงค่าของ Set Point

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.การเขียนโปรแกรมในเทอม ON-OFF



ถ้า SP มากกว่า PV สัญญาณ MV จะส่งออก 100%

ถ้า SP น้อยกว่า PV สัญญาณ MV จะส่งออก 0%

6.การเขียนโปรแกรมในส่วนของ D/R

ในส่วนของ Direct Error = SP-PV

ในส่วนของ Reverse Error = PV-SP

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นตอนการทดลอง

1.ชุดวงจร A / D Converter

หลังจากทำการต่อวงจรในส่วนของ A /D Converter เรียบร้อยแล้ว ซึ่งจะประกอบด้วยวงจร 2 ส่วน คือ วงจรในการแปลงสัญญาณกระแสไฟฟ้า 4 - 20 mA ซึ่งรับมาจากกระบวนการ แปลงสัญญาณแรงดันไฟฟ้า 0 -5V เพื่อใช้เป็นอินพุทของวงจรในส่วนของการแปลงสัญญาณแรงดันไฟฟ้าเป็นสัญญาณดิจิตอล 00 - FF อีกครั้งหนึ่ง เนื่องจากเราใช้ไอซี สำหรับการแปลงสัญญาณแรงดันไฟฟ้าเป็นสัญญาณดิจิตอลเบอร์ ADC 0801 ซึ่งชิพตัวนี้ อินพุทที่ป้อนให้ จะรับ 0 - 5 V

กระแสไฟฟ้า(mA)	แรงดันเอ๊าท์พุท(V.dc)
4	0
8	1.256
12	2.511
16	3.772
20	5.013

ตารางแสดงการปรับค่าของกระแสไฟฟ้าและแรงดันไฟฟ้าในส่วนของวงจร I /V Converter

แรงดันอินพุท(V.dc)	ดิจิตอล(HEX)
0	00
1	33
2	66
3	99
4	CC
5	FF

ตารางแสดงการปรับค่าของแรงดันไฟฟ้าและเอาท์พุทดิจิตอลในส่วนของ IC ADC 0801

หลังจากการทำการปรับค่าได้แล้ว ในส่วนของอาท์พุทดิจิตอล (00 - FF) จะเป็นินพุทดิจิตอลเพื่อเข้าสู่ MCS - 51 ทาง 8255 พอร์ต A

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.ชุดวงจร D / A Converter

จาก MCS - 51 จะส่งสัญญาณดิจิตอล ออกมาในส่วนของวงจรแปลงจากสัญญาณดิจิตอล (00 - FF) เป็นสัญญาณแรงดันไฟฟ้า (1 - 5 V.dc) โดยใช้ IC DAC 0808 ต่อร่วมเข้ากับวงจรแปลงสัญญาณ จากนั้นแรงดันอินพุท จะต่อเข้ากับวงจรแปลงสัญญาณแรงดันไฟฟ้าเป็นสัญญาณกระแสไฟฟ้า 4 - 20 mA ส่งออกไปควบคุมกระบวนการต่อไป

ดิจิตอล(HEX)	แรงดันไฟฟ้า(V.dc)
00	1.001
3F	1.998
7F	3.012
BF	4.095
FF	5.011

ตารางแสดงการปรับค่าของสัญญาณดิจิตอลและแรงดันไฟฟ้าในส่วนของวงจร D /A Converter

แรงดันไฟฟ้า(V.dc)	กระแสเอ้าท์พุท(mA)
1.006	4.000
2.001	7.995
3.002	12.010
4.002	15.996
5.001	20.000

ตารางแสดงการปรับค่าของแรงดันไฟฟ้าและกระแสไฟฟ้าในส่วนของวงจร V / I Converter

จากวงจรในส่วนของการแปลงสัญญาณดิจิตอลเป็นแรงดันไฟฟ้านั้น จะต่อมาจาก 8255 พอร์ต B ของ MCS - 51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.ชุด KEYBOARD

เนื่องจากชุดคีย์บอร์ดต่อตดยตรงกับ 8255 พอร์ต C บน และ พอร์ต C ล่าง โดยใช้ Vcc +5V ต่อเข้า MCS - 51 โดยตรง และทำการทดลองกดแต่ละคีย์ และวัดค่าฐาน 16 ที่ส่งเข้า MCS - 51 ดังนี้

คีย์ที่กด	ฐาน 2	ฐาน 16
SELECT	0000 0001	1
MODE	0000 0010	2
D/R	0000 0100	4
	0000 1000	8
	0001 0000	10
	0010 0000	20
,	0001 1000	18
,	0010 1000	28

4.ทดลองกับกระบวนการควบคุมระดับของเหลว (Level Control)

เมื่อทำการประกอบของวงจรในแต่ละส่วนประกอบเสร็จเรียบร้อยแล้ว ได้นำมาทำการทดลองควบคุมกระบวนการจริง โดยกระบวนการที่ทำการควบคุมนี้ เลือกใช้กระบวนการควบคุมระดับน้ำ (Level Control) เนื่องจากเห็นการเปลี่ยนแปลงของกระบวนการได้รวดเร็ว ตดยทดลองควบคุมกระบวนการโดยปรับค่า Gain ต่าง ๆ กัน รวมทั้งเวลา Integral Time (Ti) , Derevative Time (Td) ให้มีค่าต่าง ๆ กัน และทำการบันทึกค่าสัญญาณของ PV (Process Variable) ใน Recorder ได้ผลออกมาตาม ข้อมูลที่นำมาแสดงนี้

เนื่องจาก Action ของวาล์วควบคุมนั้น เป็นแบบ Air-to-Close ดังนั้น Action ของ Controller จึงเลือกใช้แบบ Reverse Action

จากนั้นทำการทดลองต่อไปนี้

4.1ทดลองโหมด MAN

เมื่อทำการเปิดเครื่องเรียบร้อยแล้ว หน้าจอจะแสดงสภาวะต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SP = 000.0 %			MV=000.0 %
PV = 000.0 %	HL	MAN	DIR

ทำการเลือก Action ของ Controller เป็นแบบ Reverse Action จากนั้น สังเกตุปฏิกิริยาของวาล์วควบคุม โดยเลือกค่า SP ที่ต่าง ๆ กัน

SP(%)	MV(%)	%การเปิดของวาล์ว
0	100	0
25	75	25
50	50	50
75	25	75
100	0	100

% การเปิดของวาล์ว คือ % ของปริมาณของน้ำที่ไหลผ่านวาล์วออกมา

ที่ 0% การเปิดของวาล์วคือ น้ำไม่สามารถไหลผ่านวาล์วได้

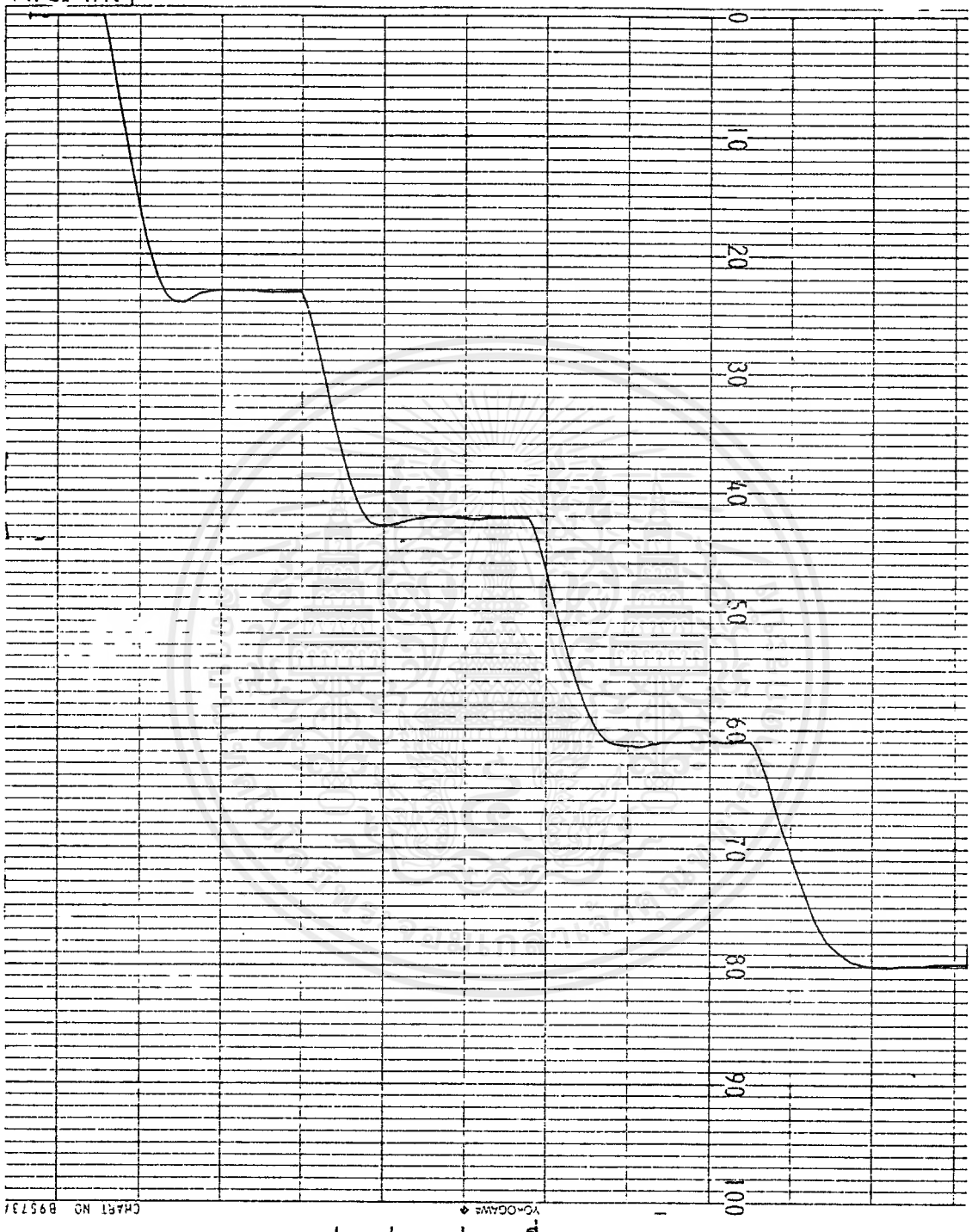
ที่ 100% การเปิดของวาล์วคือ น้ำไหลผ่านวาล์วได้ 100% เต็ม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

<u>การทดลองที่ 1</u> 1.ปรับ Speed ของ Recorder ให้เท่ากับ 2250 mm/hr

2.Set ค่า KP = 4

3.ปรับค่า SP จาก 0 ให้เป็น 20%,40%,60%,80% แล้วบันทึกค่า PVที่ SP ต่างๆ



รูปกราฟแสดงค่า PV ที่ KP = 4

ตารางแสดงค่า PV เฉลี่ยที่บันทึกได้จาก Recorder ที่ค่า SP ต่างๆ

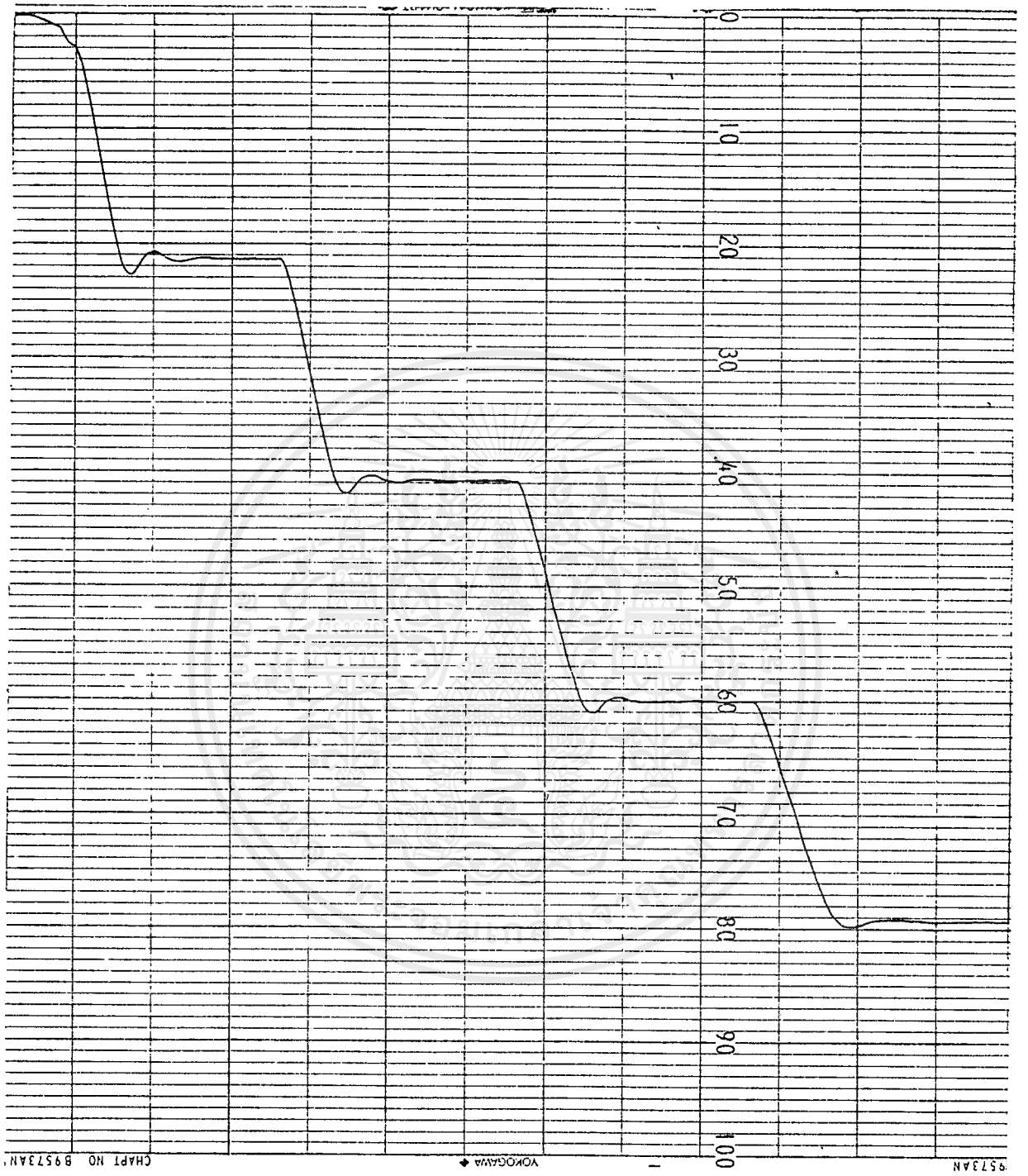
SP	20%	40%	60%	80%
ค่าPVเฉลี่ยที่ได้	21.7%	41.5%	60.3%	79%

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทดลองที่ 2 1.ปรับ Speed ของ Recorder ให้เท่ากับ 2250 mm/hr

2.Set ค่า KP = 9

3.ปรับค่า SP จาก 0 ให้เป็น 20%,40%,60%,80% แล้วบันทึกค่า PVที่ SP ต่างๆ



รูปกราฟแสดงค่า PV ที่ KP = 9

ตารางแสดงค่า PV เฉลี่ยที่บันทึกได้จาก Recorder ที่ค่า SP ต่างๆ

SP	20%	40%	60%	80%
ค่าPVเฉลี่ยที่ได้	20.7%	40%	59.3%	78.5%

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

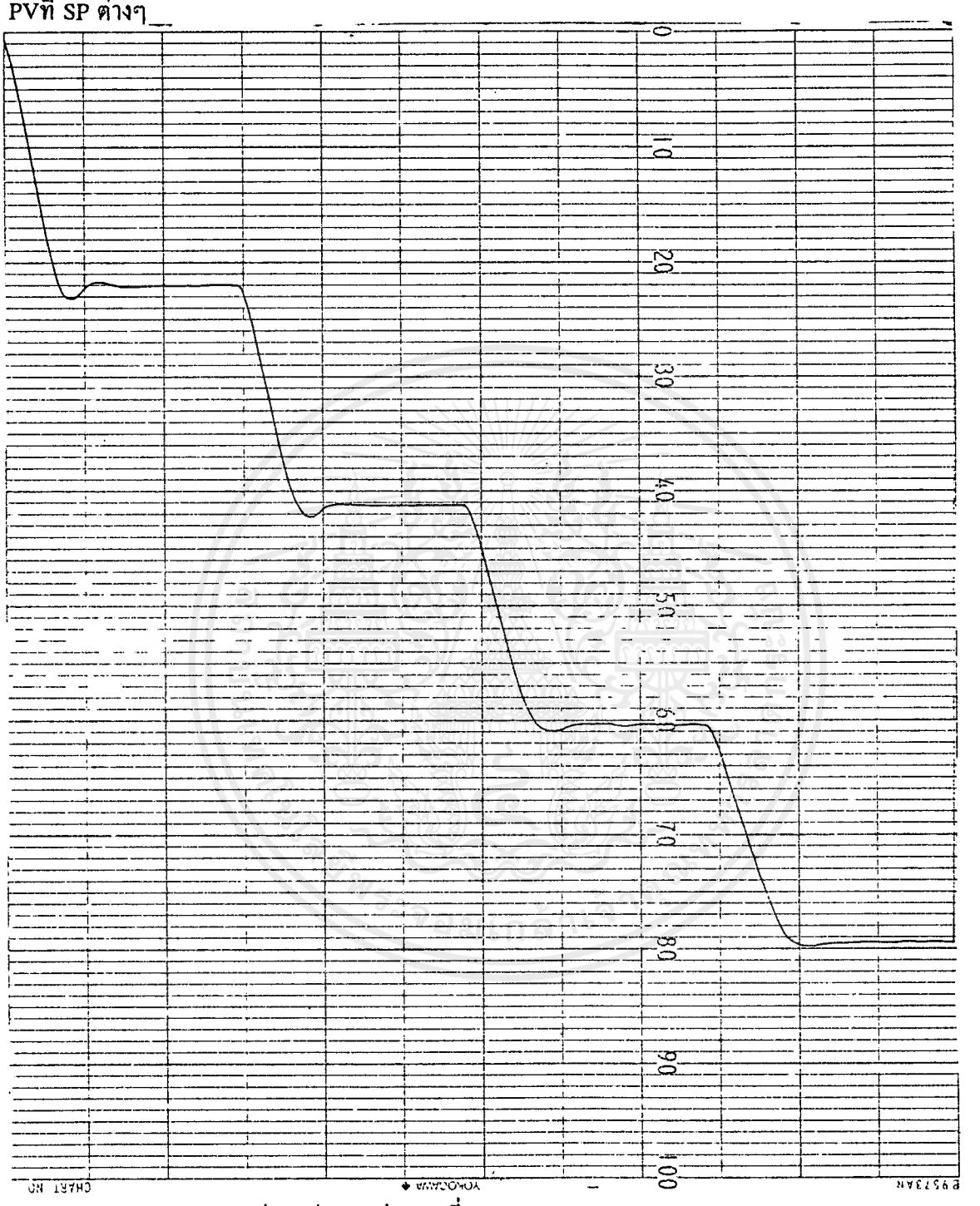
4.3 ทดลองโหมด PI

<u>การทดลองที่ 1</u> 1.ปรับ Speed ของ Recorder ให้เท่ากับ 2250 mm/hr

2.Set ค่า KP = 4 , Ti = 1438 mS

3.ปรับค่า SP จาก 0 ให้เป็น 20%,40%,60%,80% แล้วบันทึกค่า

PVที่ SP ต่างๆ



รูปกราฟแสดงค่า PV ที่ KP = 4 , Ti = 1438 mS

ตารางแสดงค่า PV เฉลี่ยที่บันทึกได้จาก Recorder ที่ค่า SP ต่างๆ

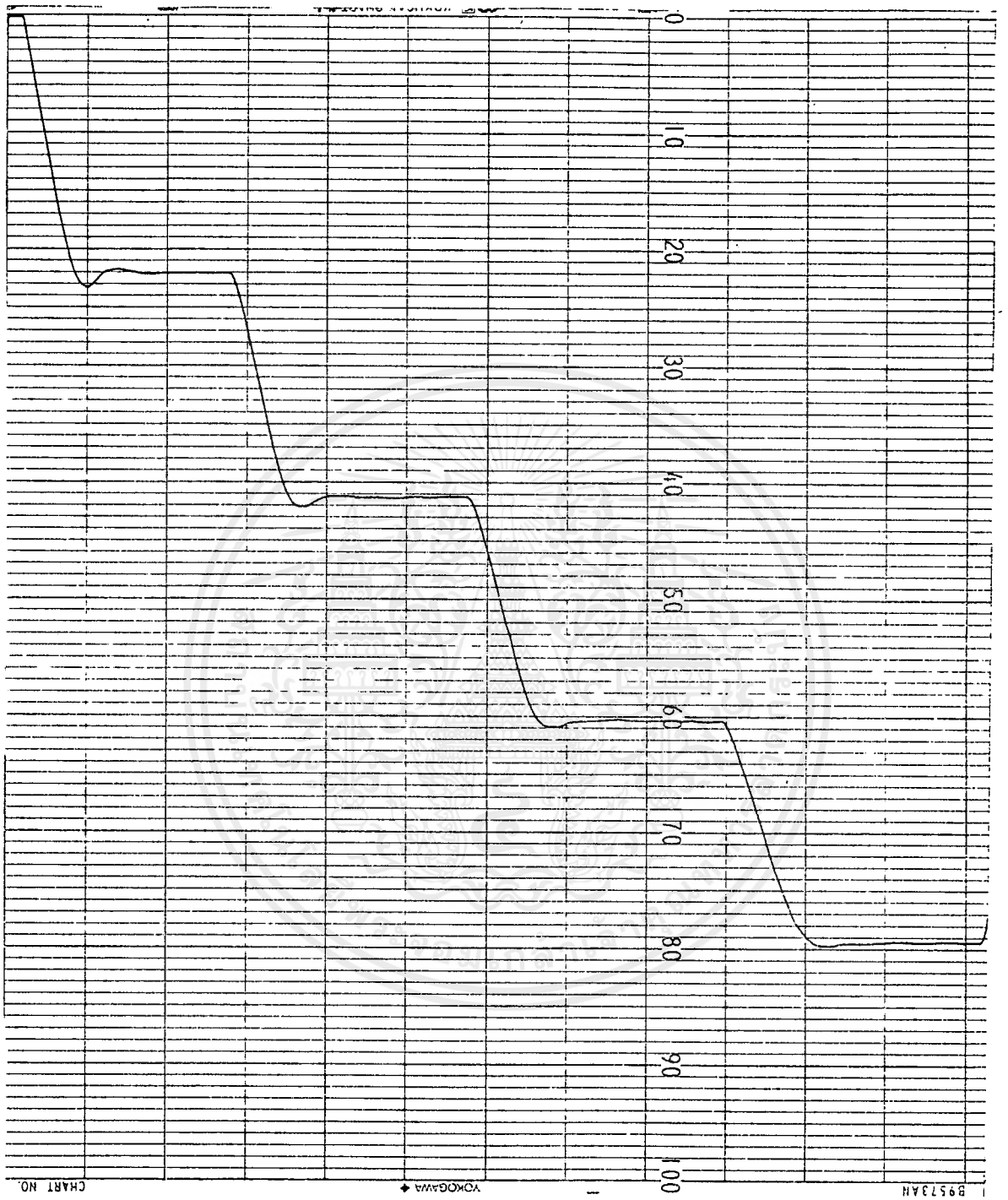
SP	20%	40%	60%	80%
ค่าPVเฉลี่ยที่ได้	21.4%	40.5%	59.6%	78.7%

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทดลองที่ 2 1.ปรับ Speed ของ Recorder ให้เท่ากับ 2250 mm/hr

2.Set ค่า KP = 4 , Ti = 722 mS

3.ปรับค่า SP จาก 0 ให้เป็น 20%,40%,60%,80% แล้วบันทึกค่า PVที่ SP ต่างๆ



รูปกราฟแสดงค่า PV ที่ KP = 4 , Ti = 722 mS

ตารางแสดงค่า PV เฉลี่ยที่บันทึกได้จาก Recorder ที่ค่า SP ต่างๆ

SP	20%	40%	60%	80%
ค่าPVเฉลี่ยที่ได้	21.4%	40.6%	59.7%	78.7%

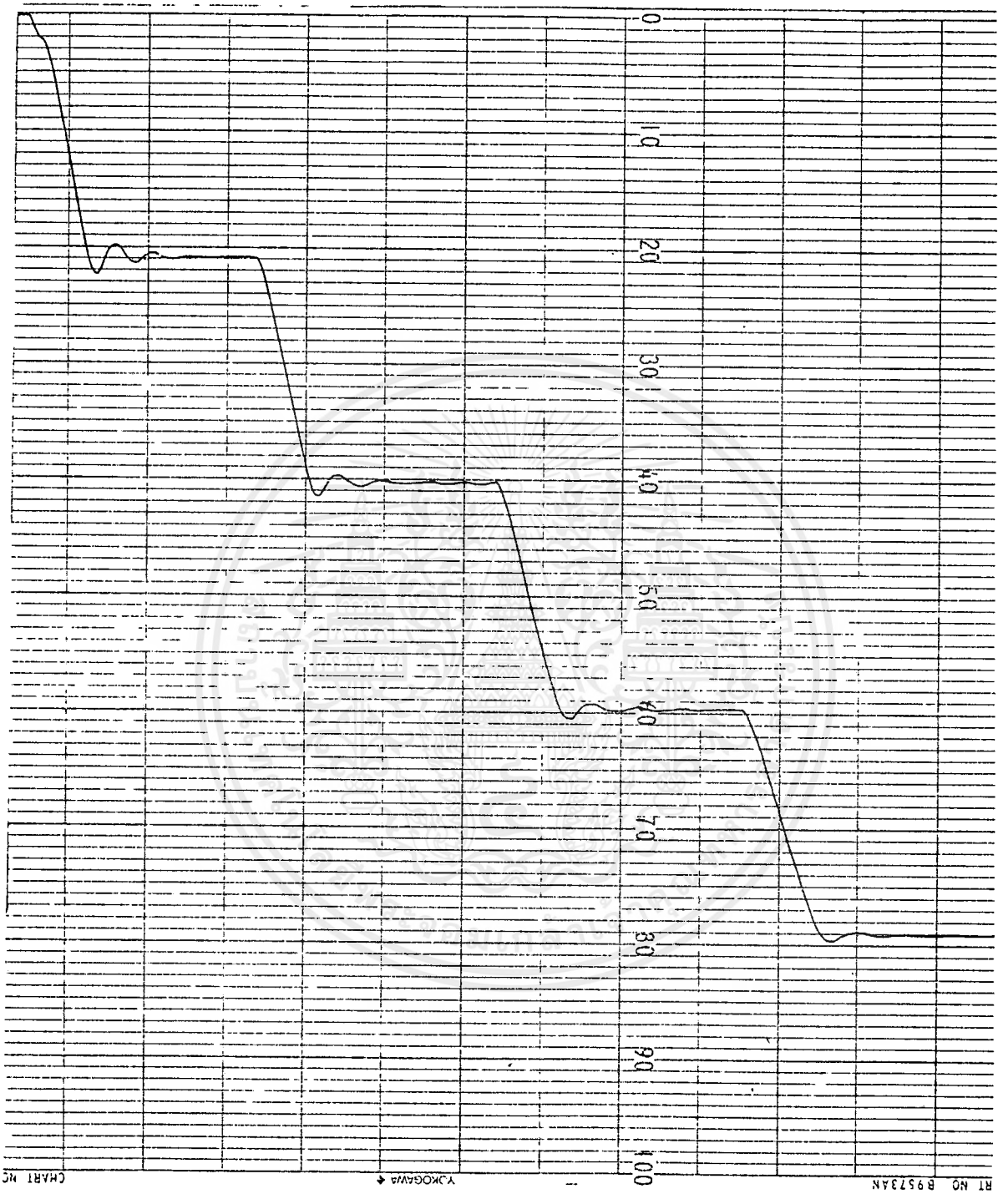
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

<u>การทดลองที่ 3</u> 1.ปรับ Speed ของ Recorder ให้เท่ากับ 2250 mm/hr

2.Set ค่า KP = 9 , Ti = 1438 mS

3.ปรับค่า SP จาก 0 ให้เป็น 20%,40%,60%,80% แล้วบันทึกค่า

PVที่ SP ต่างๆ



รูปกราฟแสดงค่า PV ที่ KP = 9 , Ti = 1438 mS

ตารางแสดงค่า PV เฉลี่ยที่บันทึกได้จาก Recorder ที่ค่า SP ต่างๆ

SP	20%	40%	60%	80%
ค่าPVเฉลี่ยที่ได้	20.3%	39.7%	59%	78.4%

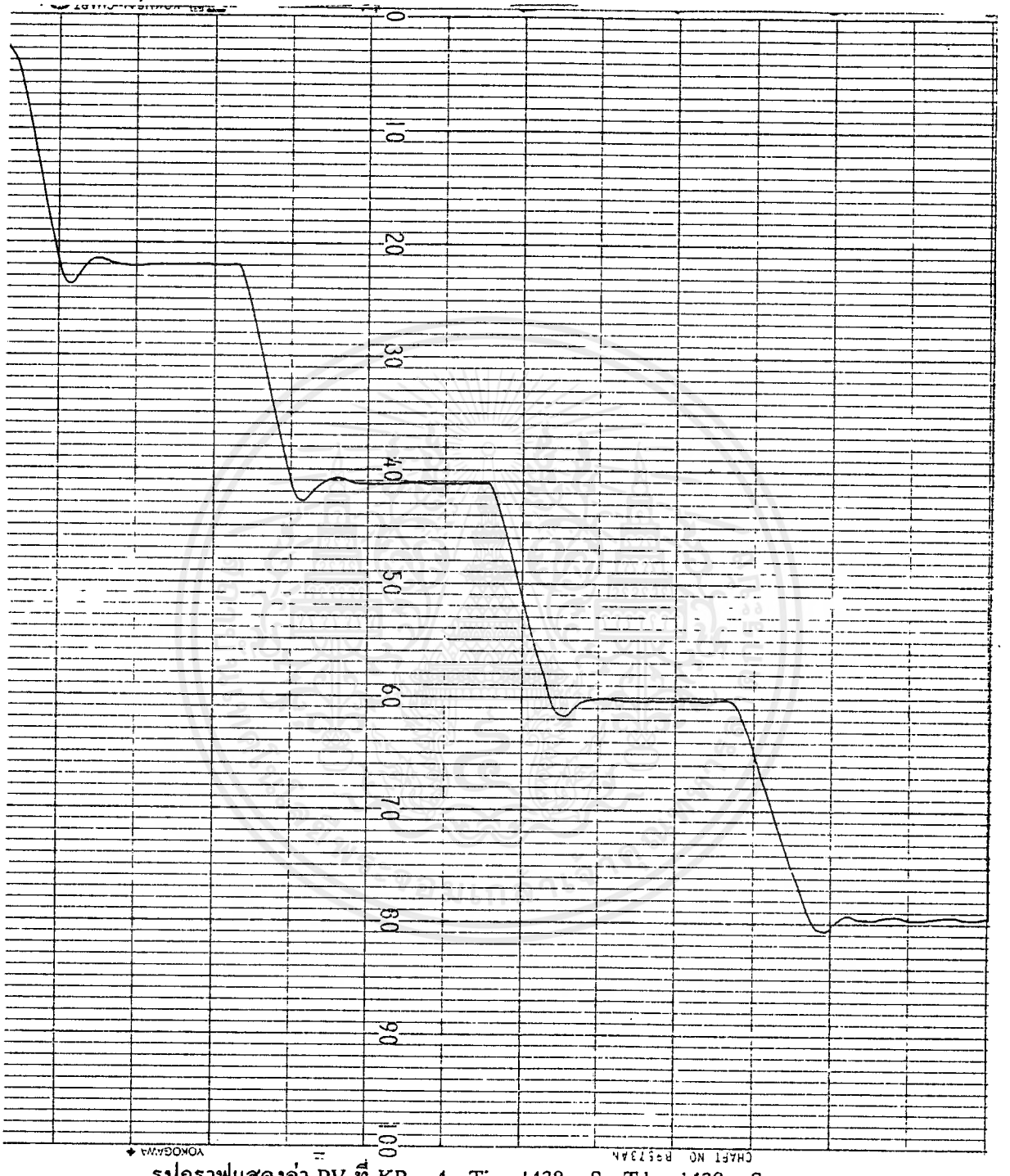
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.4 ทดลองโหมด PID

การทดลองที่ 1 1.ปรับ Speed ของ Recorder ให้เท่ากับ 2250 mm/hr

2.Set ค่า KP = 4 , Ti = 1438 mS , Td = 1438 mS

3.ปรับค่า SP จาก 0 ให้เป็น 20%,40%,60%,80% แล้วบันทึกค่า PVที่ SP ต่างๆ



รูปกราฟแสดงค่า PV ที่ KP = 4 , Ti = 1438 mS , Td = 1438 mS

ตารางแสดงค่า PV เฉลี่ยที่บันทึกได้จาก Recorder ที่ค่า SP ต่างๆ

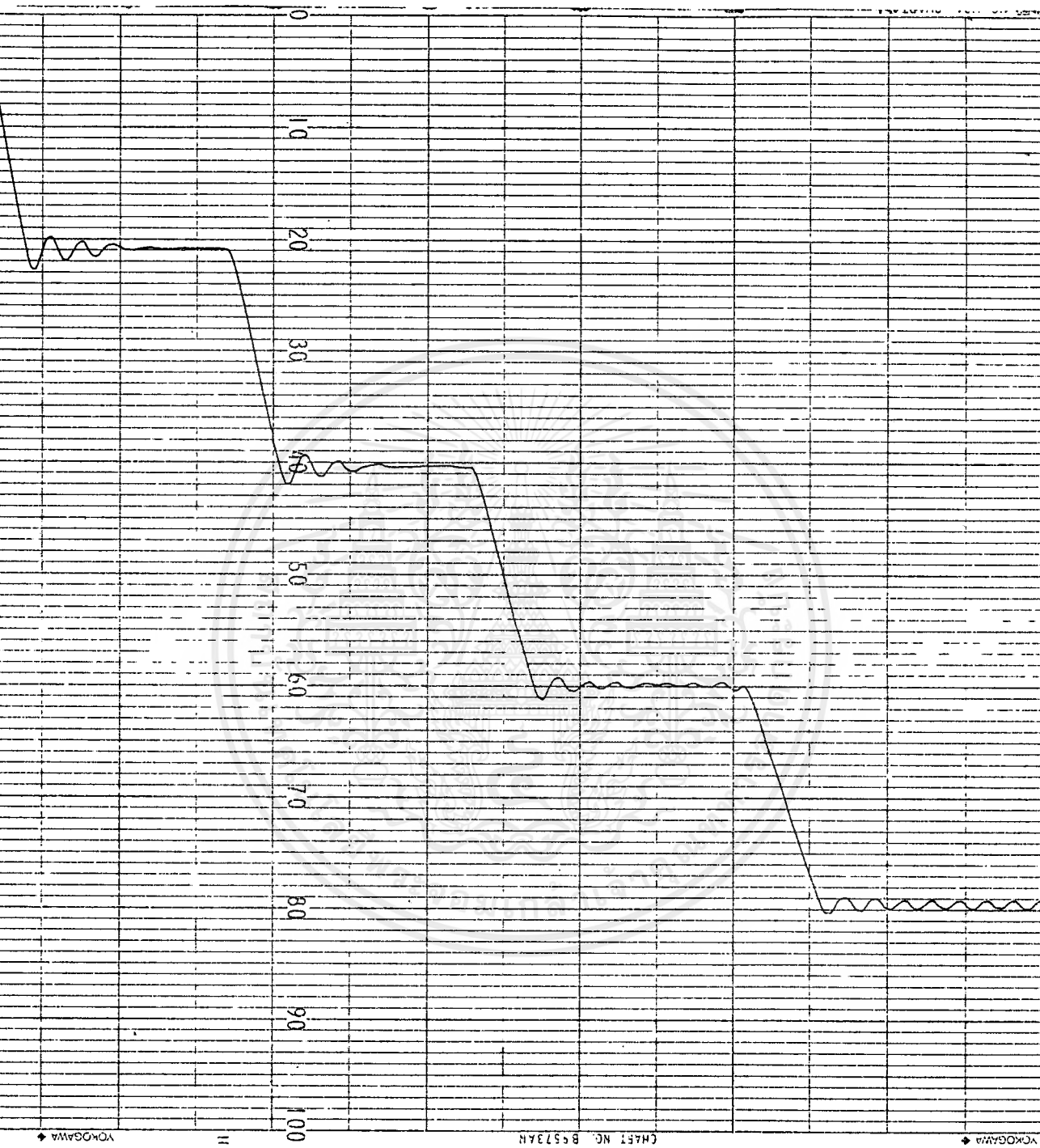
SP	20%	40%	60%	80%
ค่าPVเฉลี่ยที่ได้	21.3%	40.5%	59.6%	78.8%

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทดลองที่ 2 1.ปรับ Speed ของ Recorder ให้เท่ากับ 2250 mm/hr

2.Set ค่า KP = 9 , Ti = 1438 mS , Td = 1438 mS

3.ปรับค่า SP จาก 0 ให้เป็น 20%,40%,60%,80% แล้วบันทึกค่า PVที่ SP ต่างๆ



รูปกราฟแสดงค่า PV ที่ KP = 9 , Ti = 1438 mS , Td = 1438 mS

ตารางแสดงค่า PV เฉลี่ยที่บันทึกได้จาก Recorder ที่ค่า SP ต่างๆ

SP	20%	40%	60%	80%
ค่าPVเฉลี่ยที่ได้	20.1%	39.6%	59.3%	78.4%

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.5 การทดลองในโหมด ON -OFF

สำหรับในโหมด ON -OFF นี้ เนื่องจากเป็นแบบ Reverse Action ดังนั้น จะตรงตามเงื่อนไขนี้

เงื่อนไข	MV(%)	%การเปิดของวาล์ว
SP>PV	0	100
SP<PV	100	0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SCAN TIME

การหาค่าเวลา Scan Time ของโปรแกรมพิจารณาจากจำนวน Machine Cycle ของชุดคำสั่งทั้งหมดที่เกิดจากการวนลูปการคำนวณ หรือ ค่าของเวลาที่เครื่องควบคุมใช้ในการส่งสัญญาณออกไปควบคุมกระบวนการ 1 ครั้ง จะใช้เวลาเท่าใด พิจารณาจากสูตรการหาเวลา Scan Time การทำงานของโปรแกรม

สูตร T = (C * 12) / Crystal Frequency

เมื่อ T = เวลาที่ใช้ในการทำงานของโปรแกรม μsec

C = จำนวน Machine Cycle การทำงานของชุดคำสั่ง

Crystal Frequency = ค่าความถี่คริสตอลที่ใช้กับ 8051

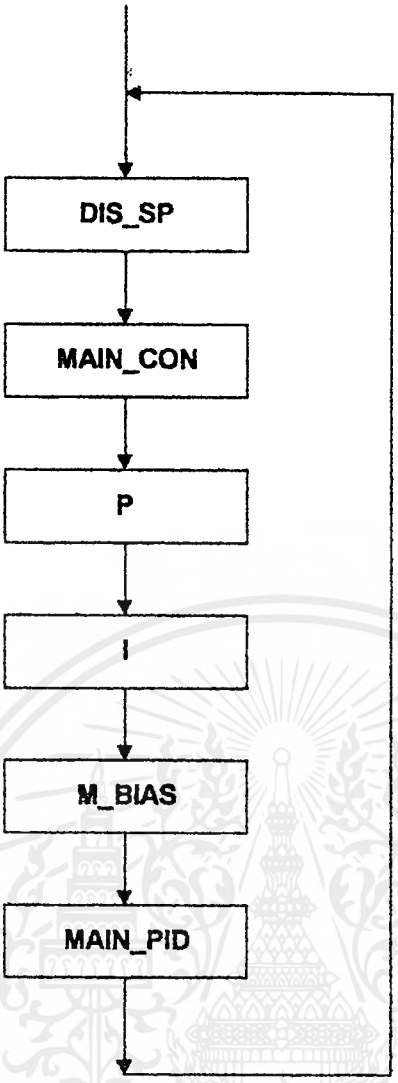
จากการพิจารณาหา Machine Cycle ของโปรแกรมได้ 5196 Machine Cycle ที่ความถี่ของ MCS-51 ที่ 11.0592 MHz

จะได้ T = (5196 * 12)/11.0592

= 5638 μsec

≈ 5.7 msec

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ลูปของเวลา Scan Time

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปัญหาที่พบและแนวทางแก้ไข

1. ปัญหาทางด้าน Hard Ware

ในส่วนของวงจรแปลงสัญญาน A/D Converter และ D/A Converter นั้น สามารถใช้งานในการทดลองได้ดี มีความเป็นลิเนียร์ที่ดี แต่เมื่อมาประกอบเข้าชุดแล้ว เมื่อทำการทดลอง บางครั้งจะพบว่ามีปัญหาในส่วนของวงจร A/D Converter คือ เมื่อ ระดับน้ำกำลังเพิ่มขึ้นถึงค่า 0% ในถังสัญญานจาก Transmitter จะส่งค่า 4 mAมาที่วงจร A/D Coverter แต่บบางครั้ง เครื่องควบคุมที่สร้างขึ้นไม่สามารถรับค่าได้ แนวทางแก้ไขคือ ใช้มิเตอร์วัดกระแสที่ไหลผ่านในส่วนของวงจร A/D Converter นั้น พบว่า ที่เครื่องควบคุมไม่รับค่านั้นเนื่องมาจาก ไม่มีกระแสไหลในวงจร ทั้งที่มีกระแสอินพุทจากกระบวนการมาแล้ว โดยสังเกตุได้ที่ Transmitter ดังนั้นจึงตรวจสอบที่ขั้ว พบว่ามีบางครั้งที่ขั้วไม่แน่น จึงทำการแก้ไขเรียบร้อยแล้ว

ปัญหาต่อมาที่สำคัญก็คือ การจ่ายโหลดของเครื่องควบคุมที่สร้างขึ้นมานั้น สามารถจ่ายโหลดไปที่ตัว I/P Converter ได้อย่างเพียงพอ คือสามารถทำการควบคุม Control Valve ให้มีการเปิด-ปิดได้ แต่เมื่อมีการนำ Recorder มาต่อเพื่อทำการบันทึกค่า จะไม่สามารถควบคุมได้ เนื่องมาจากกระแสถูกแบ่ง ซึ่งสามารถวิเคราะห์ได้คือ ที่ตัว I/P Converter จะมีค่าความต้านทานของตัวมันคือ ที่ 250 โอห์ม ซึ่งตัวคอนโทรลเลอร์สามารถจ่ายกระแสได้พอ แต่ที่ Reccorder ต้องใช้ความต้านทาน 250 โอห์ม มาต่อเพื่อเปลี่ยนเป็นแรงดันไฟฟ้า 1-5 V.dc. ซึ่งโหลดที่เพิ่มขึ้นมาอีก 250 โอห์มนั้นทำให้เครื่องควบคุมที่สร้างขึ้นมาจ่ายกระแสไม่พอ แนวทางแก้ไขก็คือ ต้องปรับปรุงในส่วนการจ่ายกระแสของวงจร V/I Converter ให้จ่ายกระแสทนโหลดได้สูงขึ้น หรืออีกวิธีหนึ่งก็คือ ใช้ความต้านทานที่ต่อที่ Recorder ลดลง แล้วทำการปรับที่ตัว Reccorder ในการบันทึกค่า

2. ปัญหาทางด้าน Soft Ware

เนื่องจากข้อจำกัดทางด้านการคำนวณของ MCS-51 ที่มีคำสั่งที่รองรับการคำนวณทางคณิตศาสตร์อยู่ 4 คำสั่ง คือ คำสั่งในการบวก ADD,การลบ SUBB, การคูณ MUL และการหาร DIV ทำให้ในการคำนวณที่ต้องใช้การคำนวณมาก อาจจะมีความยุ่งยากพอสมควร

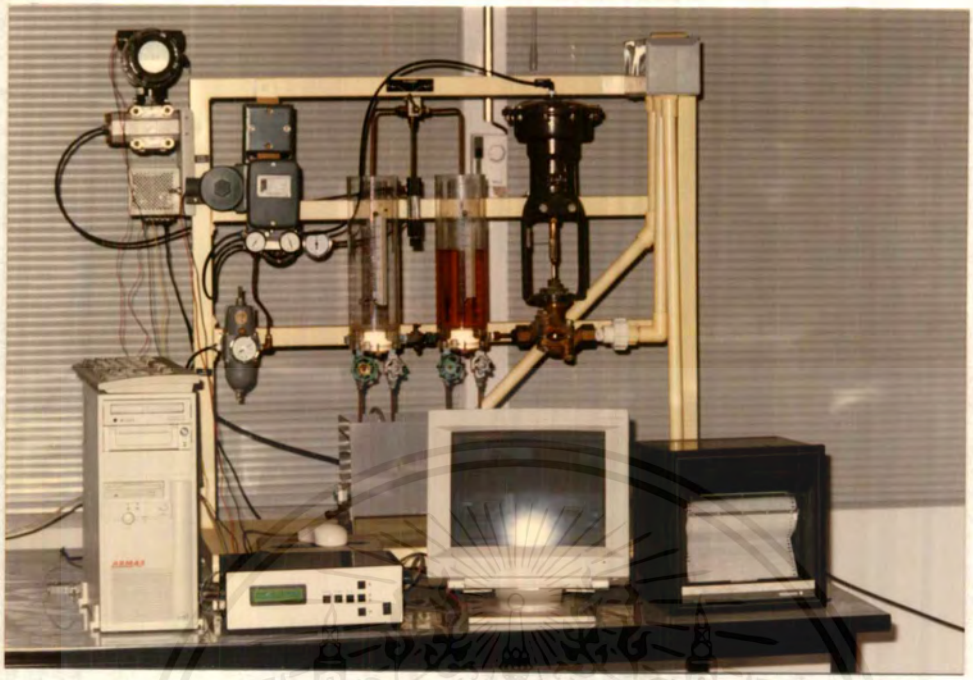
ในการคำนวณคำสั่ง เช่น การอินทริเกรต และ การดิฟเฟอเรนเชียล นั้น จะต้องทำคำสั่งเหล่านี้ให้อยู่ในรูปของการบวก การลบ การคูณ และ การหารเท่านั้น จึงเป็นการยากที่จะเขียนโปรแกรมทำงาน

และข้อจำกัดอีกบางประการคือ ที่ตัว MCS-51 มีจำนวนบิตในการคำนวณอยู่ 8 บิต ดังนั้นความละเอียดในการคำนวณจะอยู่ที่ 256 ค่า (00-FF) ไม่เหมือนกับเครื่องคอนโทรลเลอร์ทั่วๆไป ที่

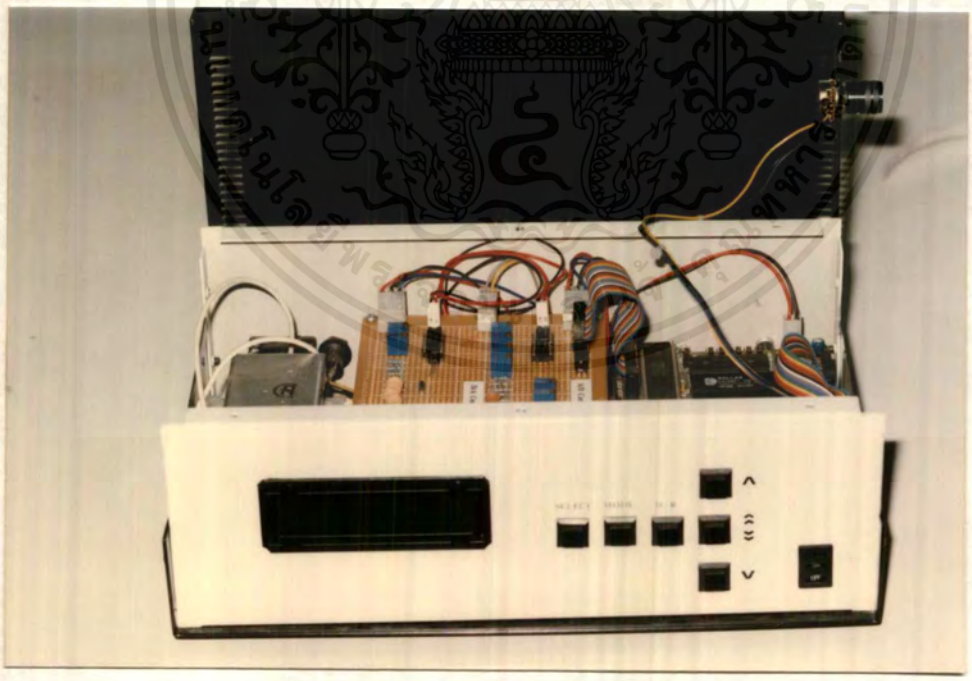
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มีความละเอียดในการคำนวณอยู่ที่ 10 บิต หรือ 12 บิต ดังนั้นในการทำ PID Controller นี้ จึงเป็นขั้นศึกษาการเขียนโปรแกรมควบคุมเท่านั้น ซึ่งในทางปฏิบัติก็สามารถทำการควบคุมได้ดี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

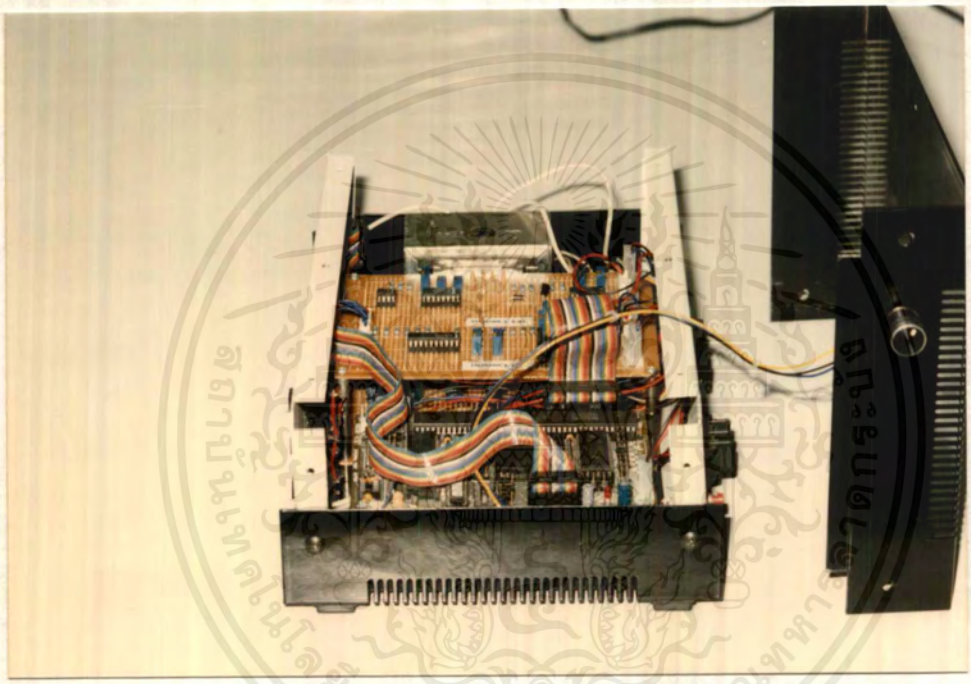


รูปแสดง UNIT 02 (PID CONTROLLER) ทำการต่อควบคุมกระบวนการระดับน้ำ



รูปแสดง UNIT 02 (PID CONTROLLER)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปแสดงการต่อวงจรภายในของ UNIT 02

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปแสดงหน้าจอของ ST.No. 02

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 7

COMPUTER PROCESS CONTROL

เมื่อได้มีการพัฒนาทางอิเลคโทรนิคส์มาถึงยุคของ Microprocessor จึงทำให้เกิดการใช้ Digital Processing Unit (หรือ Microprocessor Controller) ซึ่งจะประกอบด้วย Microprocessor , หน่วยความจำ , หน่วยแปลงสัญญาณ และหน่วยรับข้อมูล ด้วยการใช้ Microprocessor Controller หลายๆ ชุดที่แยกอิสระกันทำงาน จะสามารถควบคุมทั้ง plant ได้ การเชื่อมโยงติดต่อระหว่าง Controller ใช้หลักการส่งสัญญาณ serial digital และติดต่อกับ Master Computer หรือ Operater CRT Console ได้(ถ้าต้องการ) Operater CRT Console โดยทั่วๆ ไปจะใช้ Microprocessor ภายในตัวมันเอง โดยไม่ต้องอาศัย computer ภายนอกมาควบคุม อีกทั้งยังสามารถทำงานคล้าย computer ได้ เช่น แสดงค่าต่างๆ ออกมาในรูปกราฟ , สั่งงาน printer , เก็บข้อมูล ฯลฯ ในกรณีที่ Operater Console ไม่อาจใช้งานได้ ก็ไม่มีผลกระทบต่อการทำงานของ Microprocessor Controller ซึ่งมีการควบคุมแบบ PID โดยอัตโนมัติ

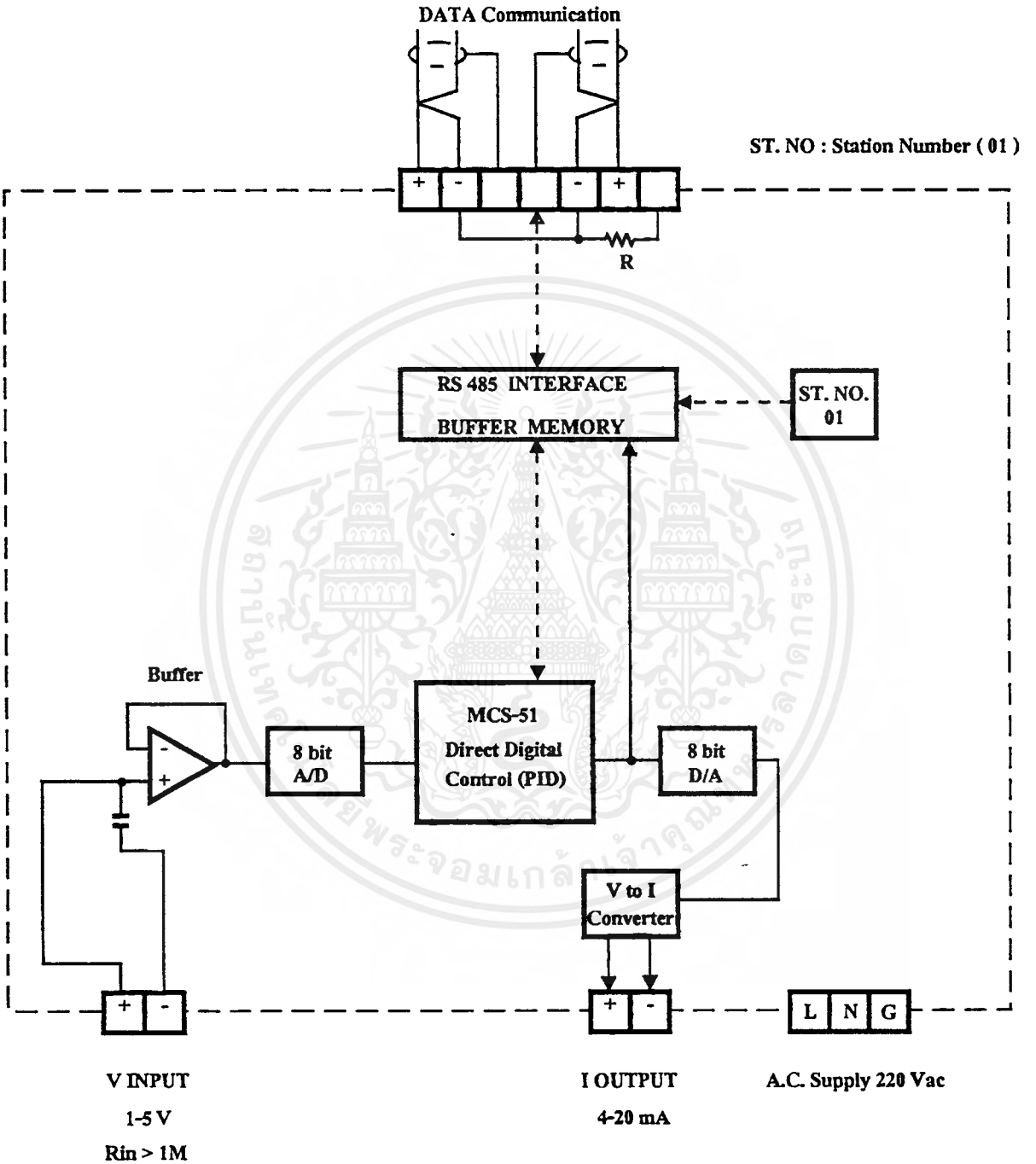
มีข้อได้เปรียบมากมายสำหรับ Microprocessor - Based Distributed Control System ในระบบที่ซับซ้อนมากๆ การติดตั้งและเดินสายอุปกรณ์ของ Distributed Control จะถูกกว่า Analog System มาก นอกจากนี้เนื้อที่สำหรับห้องควบคุมก็ใช้น้อยกว่า เพราะว่าสามารถกระจาย Microprocessor Controller ไปใน Field ได้ จำนวนสายไฟและสายสัญญาณก็ลดลงไปได้มาก การเพิ่ม process computer ก็ทำได้ง่ายโดยการพ่วงเข้าไปใน Buses หรือ Data Highway สามารถเรียก Process variable และ Control output จาก Microprocessor Controller แต่ละตัวได้ นอกจากนี้ยังอาจใช้ computer เป็นตัวทำ Process Optimization ได้อีกด้วย แต่อย่างไรก็ตาม การใช้ Micro computer ในการควบคุมระบบโดยตรงก็เป็นสิ่งที่กระทำได้และอาจเหมาะสมกับบางโปรเซสด้วย

จากพื้นฐานในบทที่ได้กล่าวมาแล้วนั้น เราจะพบลู่ทางในการประยุกต์ใช้ Computer เข้าควบคุมโปรเซสได้ ซึ่งผลที่เราต้องการจากการใช้ Computer ในการควบคุมนี้คือ ให้เกิดความแม่นยำ ความสามารถในการคำนวนที่ซับซ้อน ความสะดวกในการดัดแปลงโครงสร้างของระบบควบคุม ความสามารถในการจัดการ หรือต้องการจัดเก็บข้อมูลจำนวนมากๆ เป็นต้น

ในการนำ Computer มาใช้ในงานควบคุมนี้ สามารถแบ่งได้เป็น 2 ประเภทใหญ่ๆ คือ SCC และ DDC ซึ่งคือสิ่งที่เราพัฒนาและประยุกต์ใช้โดยมีรายละเอียดดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

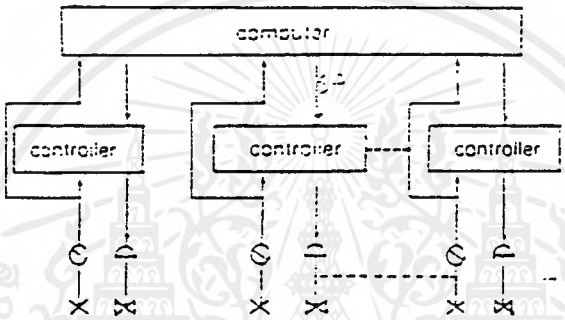
Analog Input/Output Controller (PID)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7.1. Supervisory Computer Control (SCC)

ในการใช้คอมพิวเตอร์เป็นตัวกำหนดค่าเป้าหมาย (Setpoint) ให้ตัวควบคุม (Controller) ซึ่งเรียกได้ว่าเป็นการควบคุมแบบตรวจตรา (Supervisory) เพราะเราสามารถเฝ้าดูการเปลี่ยนแปลงของโปรเซสแต่ละตัวได้ เหมาะสำหรับใช้ในงานที่ตัวควบคุม (Controller) หลายๆ ชุด การที่เราจะควบคุม Controller หลายๆ ตัวในโรงงานด้วยตัวเองนั้นเป็นเรื่องที่ยุ่งยาก เราก็จะใช้ระบบ SCC นี้ คอยตรวจตราโปรเซสต่างๆ ผ่าน Controller ประจำโปรเซสนั้นๆ ได้ พิจารณาดังรูป



Supervisory Computer control System (SCC)

รูปที่ 7.1 แสดงการใช้ Supervisory Computer Control (SCC)

จุดประสงค์หลักอันดับหนึ่งคือ ต้องการส่งค่า Setpoint ให้กับ Controller หมายเลขใดหมายเลขหนึ่งตามแต่จะต้องการ โดยที่ Controller ตัวอื่นๆ จะไม่ได้รับการติดต่อส่งค่า ถ้าไม่ใช่ตัวที่ต้องการ เราสามารถแสดงรายละเอียดการต่อ Computer ร่วมกับ Process และ Controller ได้ดังรูปต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

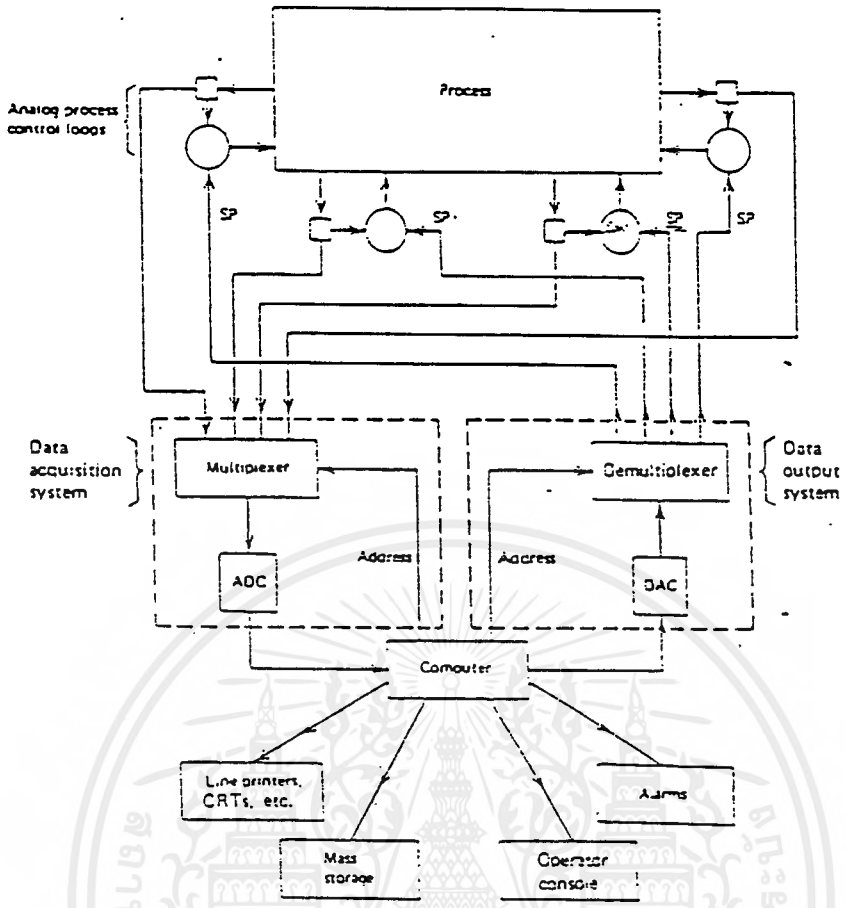


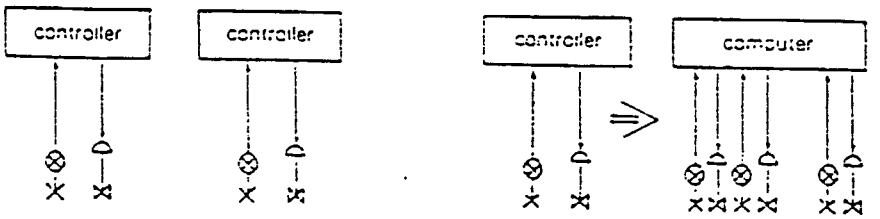
FIGURE 10.6 In computer supervisory control the computer sets the setpoints analog process-control loops.

รูปที่ 7.2 การต่อ Computer ร่วมกับ process

7.2.Direct Digital Control (DDC)

คือการนำเอา Computer มาใช้ทำงานควบคุมแทนตัวควบคุม (Controller) เสียเอง การทำงานในลักษณะนี้จะกระทำได้โดยเขียนสมการการควบคุมที่เหมาะสมไว้ใน Computer เพื่อให้มันทำการคำนวนหาสัญญาณการควบคุมที่เหมาะสมออกมา ซึ่งสมการการควบคุมนี้มักจะใช้กฎเกณฑ์การควบคุมชนิด PID เป็นส่วนใหญ่ นั่นคือเราจะต้องสร้างโปรแกรมเพื่อให้ Computer ทำการ Simulate ตัวเองให้เหมือนกับเป็น PID Controller ลักษณะของการใช้งาน DDC จะกระทำได้ดังรูปที่ 7.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 1 Direct Digital Control System (DDC)

รูปที่ 7.3 DDC

สำหรับการต่อ Computer เข้าร่วมกับ process แสดงได้ดังรูปที่ 7.4 และในบทที่กล่าวมาแล้วนั้น เราได้กล่าวถึง PID Controller มาแล้ว ดังนั้นการประยุกต์ใช้งานโดยใช้ซอฟแวร์เขียนเลียนแบบการควบคุม PID ที่กล่าวมาแล้วก็สามารถทำได้โดยง่าย โดยเปลี่ยนสมการเหล่านั้นให้อยู่ในรูปของอัลกอริธึ่มที่เครื่องคอมพิวเตอร์สามารถเข้าใจได้ ดังจะกล่าวต่อไปนี้

DDC Algorithm of Control Action

ในการคำนวณสมการต่างๆ นั้น จะต้องทราบค่าความผิดพลาดของกระบวนการที่เราทำการควบคุมอยู่เสียก่อน โดยในการคำนวณค่าผิดพลาดนั้นจะแบ่งเป็น2แบบ ดังนี้คือ

1.A. direct - acting controller ซึ่งสัญญาณการควบคุมของตัวควบคุมแบบนี้ จะมีค่าเพิ่มขึ้นเมื่อค่า Process variable มีค่าเพิ่มขึ้น ซึ่งตัวอย่างของกระบวนการที่จะควบคุมเช่น ระบบทำความเย็น เป็นต้น

สมการในการคำนวณ

$$e_{direct} = PV - SP$$

2.An. Inverse - acting controller ซึ่งสัญญาณการควบคุมของตัวควบคุมแบบนี้ จะมีค่าลดลงเมื่อค่าของ Process variable มีค่าเพิ่มขึ้น ซึ่งตัวอย่างของกระบวนการที่จะควบคุมเช่น ระบบทำความร้อน เป็นต้น

สมการในการคำนวณ

$$e_{inverse} = SP - PV$$

จากที่ได้กล่าวมาแล้วในบทก่อนๆ การควบคุมแบบ PID นั้นจะมีสัญญาณควบคุม (ซึ่งตัวควบคุมได้ส่งออกมา) เป็นในลักษณะดังนี้

$$Vo = Kpe + Ki\int edt + Kd\frac{de}{dt}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

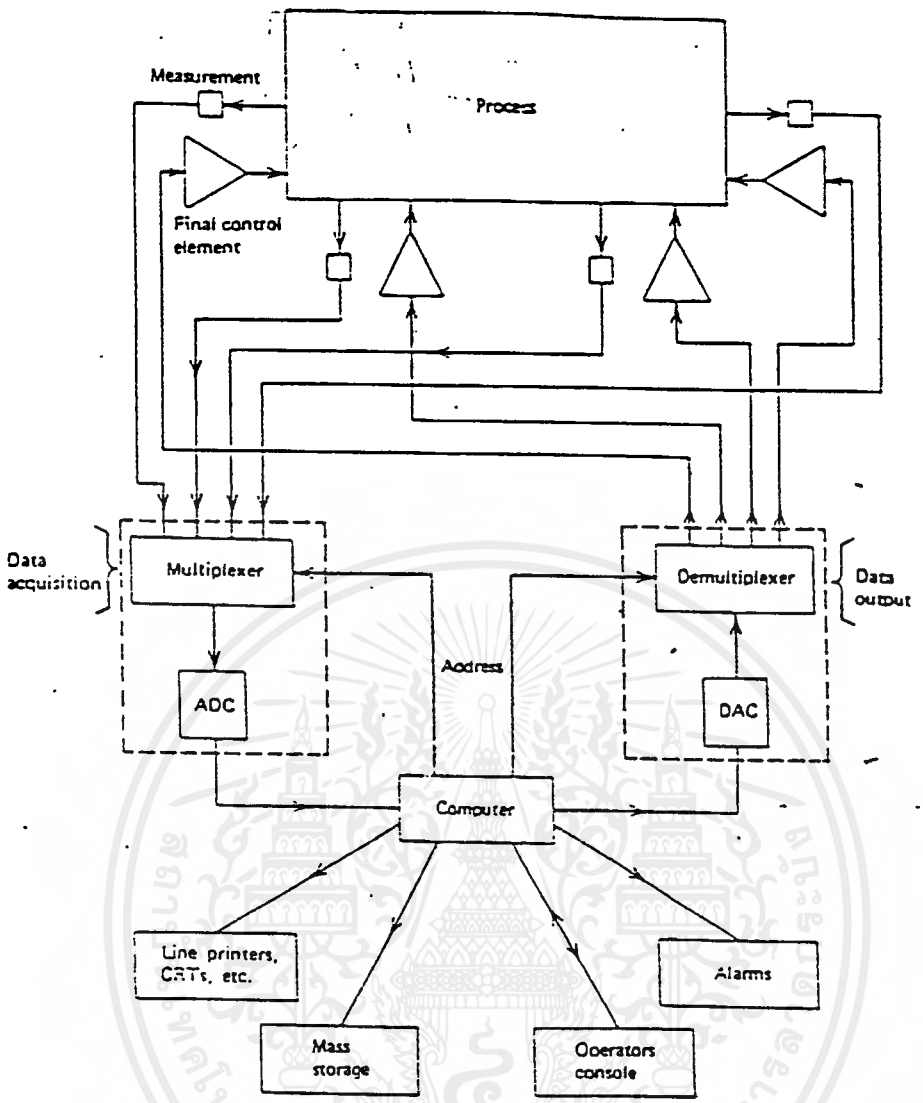


FIGURE 10.9 In direct digital control (DDC) the analog process-control loop is replaced by digital computer operations.

รูปที่ 7.4 การต่อคอมพิวเตอร์กับโพรเซสในแบบDDC

ซึ่ง e = ค่าผิดพลาด

Kp, Ki, Kd = ค่าคงที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถ้าตั้งค่า Kd = 0 จะทำให้เทอมของ Derivative ไม่ทำงาน และถ้าเราตั้งค่า Ki = 0 อีก ก็จะทำให้เทอมของ Integralไม่ทำงาน ซึ่งจะเหลือแต่เทอมของ Proportional เท่านั้น ค่าคงที่ทั้ง 3 ตัวนี้ จะมีผลต่อคุณภาพของการควบคุม

ในการทำให้ Microcomputer สามารถคำนวณสมการดิฟเฟอร์เรนเชี่ยลแบบต่อเนื่องได้นั้น จะต้องแปลงสมการดังกล่าว ให้เป็นแบบ discrete difference equation เสียก่อน ดังนี้

ทำการดิฟเฟอร์เรนทิเอททั้ง 2 ข้าง

$$\frac{dVo}{dt} = Kp\frac{de}{dt} + Ki\frac{d}{dt}\left(\int edt\right) + Kd\frac{d^2e}{dt^2}$$

$$\frac{dVo}{dt} = Kp\frac{de}{dt} + Kie + Kd\frac{d}{dt}\left(\frac{de}{dt}\right)$$

สมการนี้จะบอกเราว่า output จะเปลี่ยนแปลงไปอย่างไรในช่วงเวลา (dt) แต่ในระบบ Microcomputer - Based System นั้น The cycle time (T) จะใช้เป็น (dt) ซึ่งจะสามารถเขียนสมการได้ใหม่ดังนี้

$$\frac{\Delta Vo}{T} = Kp\frac{\Delta e}{T} + Kie + Kd\frac{\Delta}{T}\left(\frac{\Delta e}{T}\right)$$

ซึ่ง T เป็น The cycle time แล้วคูณด้วย T ตลอดสมการ จะได้ดังนี้คือ

$$\Delta Vo = Kp\Delta e + KieT + Kd\Delta\left(\frac{\Delta e}{T}\right)$$

เปลี่ยนค่า Vo , ΔVo ให้เป็นผลต่างระหว่างค่าสัญญาณควบคุมของการสุ่มครั้งที่ n และ ค่าสัญญาณควบคุมของการสุ่มครั้งก่อน

$$\Delta Vo = V_{on} - V_{on-1}$$

และค่าผิดพลาดก็ทำให้เป็นผลต่างด้วยเช่นกัน ดังนี้

$$\Delta e = e_n - e_{n-1}$$

ทำการเขียนสมการใหม่ได้ดังนี้

$$Vo - V_{on-1} = Kp\left(e_n - e_{n-1}\right) + KieT + \frac{Kd}{T}\left(\Delta e_n - \Delta e_{n-1}\right)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทำการกระจายเทอมสุดท้าย โดยใช้สมการดังนี้

$$\Delta e_n = e_n - e_{n-1}$$

$$\Delta e_{n-1} = e_{n-1} - e_{n-2}$$

เขียนสมการใหม่ได้ดังนี้คือ

$$Vo - V_{on-1} = Kp(e_n - e_{n-1}) + Kie_nT + \frac{Kd}{T}\left[(e_n - e_{n-1}) - (e_{n-1} - e_{n-2})\right]$$

หรือ

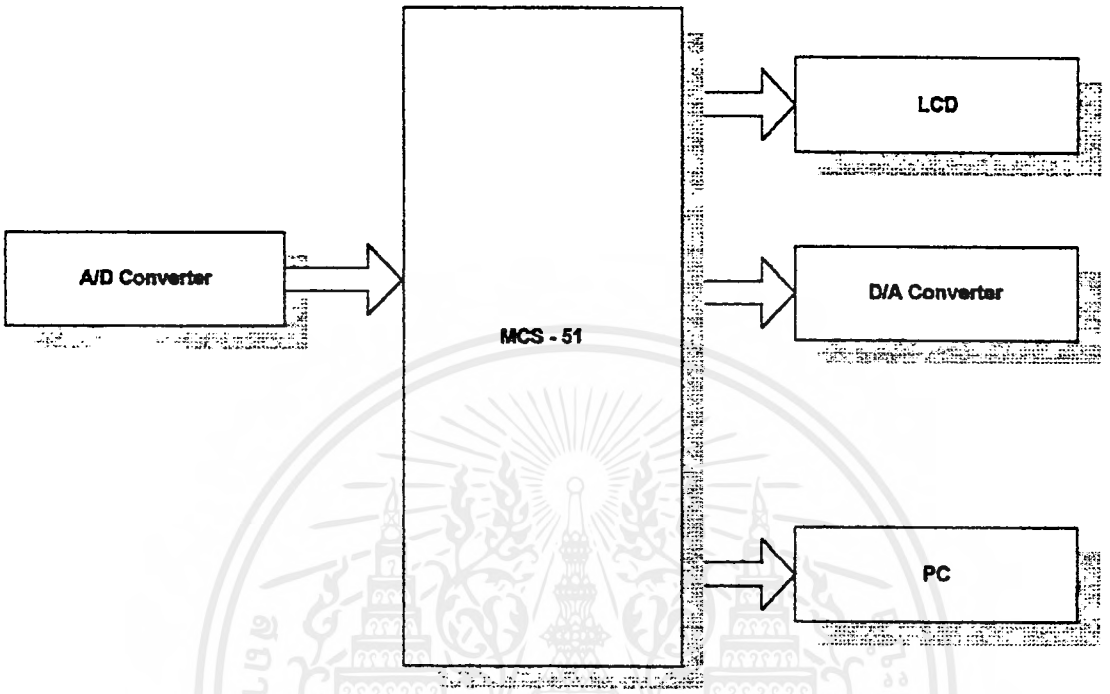
$$Vo - V_{on-1} = Kp(e_n - e_{n-1}) + Kie_nT + \frac{Kd}{T}\left[(e_n - 2e_{n-1} + e_{n-2})\right]$$

และในที่สุดก็ได้สมการดังนี้

$$\boxed{Vo - V_{on-1} = Kp(e_n - e_{n-1}) + Kie_nT + \frac{Kd}{T}\left[(e_n - 2e_{n-1} + e_{n-2})\right]}$$

โดยนำสมการที่ได้นี้ เอาไปเขียนใน Computer เพื่อให้สามารถทำการควบคุม process ได้ตามที่เราต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

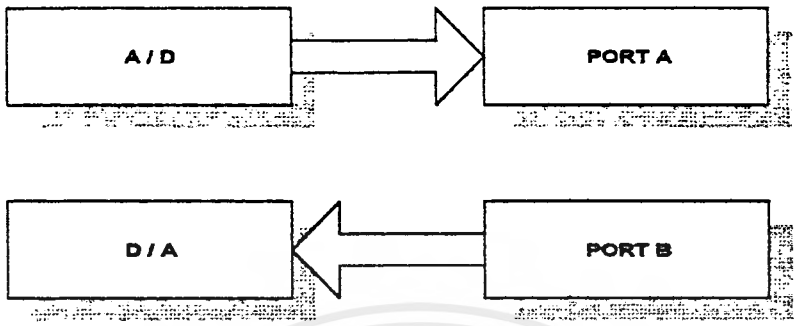


รูปแสดงส่วนประกอบทางด้าน Hard Ware

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.MCS-51

1.1 มี Input/Output Port 8255 2 พอร์ต



Address F800H + 8255 offset address = Actual address

Port A ตำแหน่งแอดเดรส F800H + 00H = F800H

Port B ตำแหน่งแอดเดรส F800H + 01H = F801H

Mode Port ตำแหน่งแอดเดรส F800H + 03H = F803H

1.2 หน่วยความจำ Program and Data MeMory ขนาด 8-32 Kbyte

1.3 มีพอร์ทของ LCD ให้มาต่อได้เลย

1.4 มีพอร์ทสื่อสารอนุกรม RS 232 ให้มาด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. A/D Converter

- I/V Convertor (4-20 mA to 0-5 V.dc)
- A/D Converter (0-5 V.dc to 00-FF)

3. D/A Converter

- D/A Converter (00-FF to 1-5 V.dc)
- V/I Converter (1-5 V.dc to 4-20 mA)

4. LCD

-ใช้ LCD แบบ Dot Matrix เบอร์ DMC 202

5. PC

-ส่งออก PC โดยผ่านทาง RS 232

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คุณสมบัติของโปรแกรม ส่วน Analog I/O Controller (PID)

1. เป็น PID Controller ที่ประมวลผลโดย คอมพิวเตอร์เป็นหลัก จึงสะดวกในการคำนวณที่ซับซ้อน สะดวกต่อการเปลี่ยนแปลง และ สามารถเพิ่มคุณสมบัติ ต่างๆ ได้มาก

2. ส่วน Input Output ใช้ไมโครคอนโทรลเลอร์ ตระกูล MCS-51 ทำให้สะดวกต่อการเรียนรู้ ทำความเข้าใจ

3. สามารถ แสดงผล PV และ MV เป็น กราฟ แบบ เรียลไทม์ (สามารถ Zoom ได้) ค่าเกณฑ์ P ,I ,D ,Bias และ Setpoint สามารถป้อนได้โดยตรงที่คอมพิวเตอร์

4. สามารถทำงานในแบบ Manual และ Auto โดย การสลับระหว่างสองโหมด เป็นแบบ Bumpless transfer สามารถ กลับ Action ได้

5. สามารถ แสดง Cycle time ในแต่ละรอบได้ ที่หน้าจอ

6. มี สัญญาน เตือน PV ที่สูงหรือต่ำเกินกว่าที่กำหนด

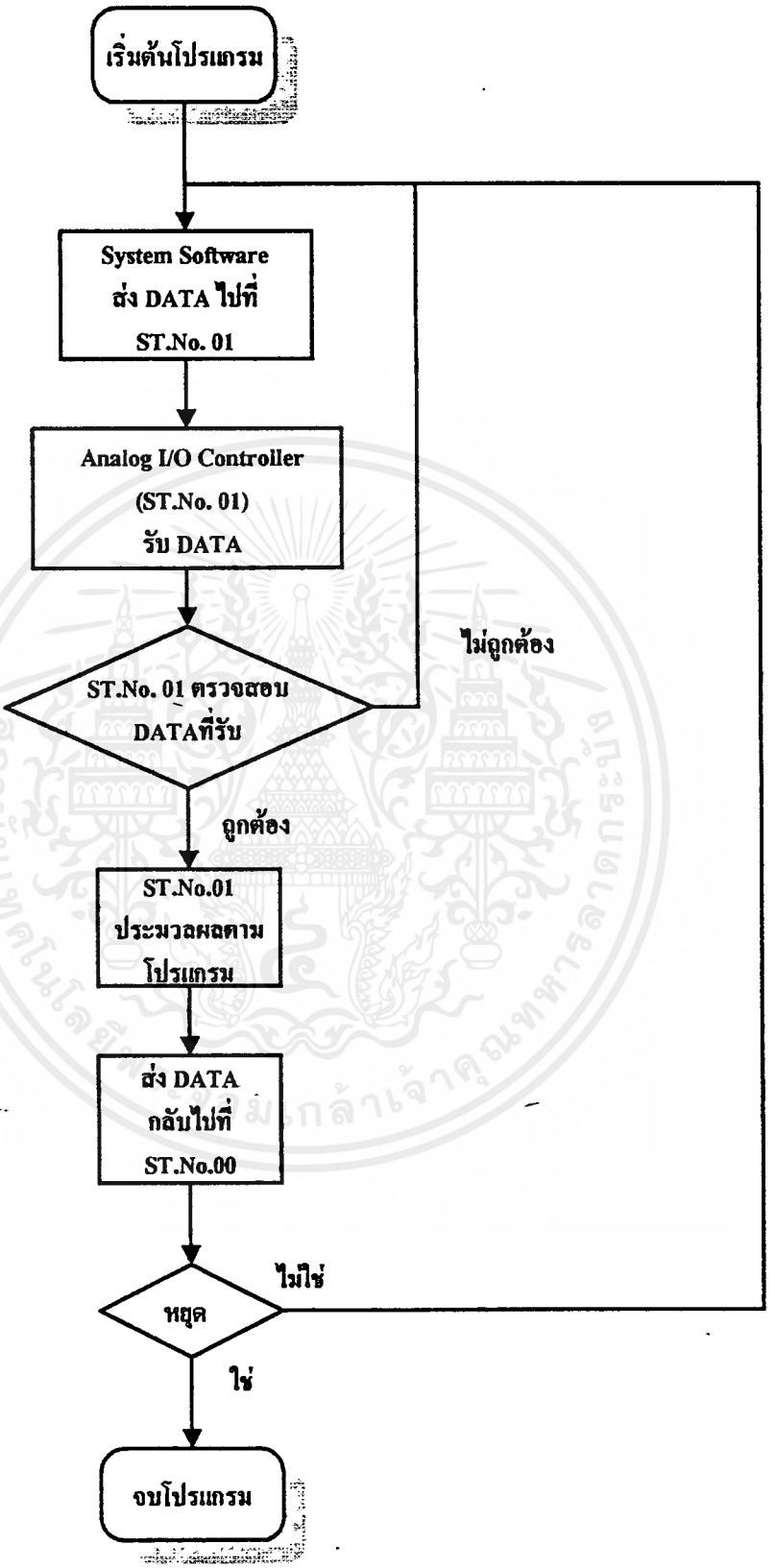
7. สามารถแสดง ข้อมูลตาม Protocol ที่ส่งสื่อสารกันบนหน้าจอทั้งรับและส่งแบบ เวลาจริง ทำให้สามารถตรวจสอบได้ง่าย

8. มีการตรวจสอบความถูกต้องของการสื่อสาร และแสดงผลบนหน้าจอ

9. มีโหมดการตรวจสอบโดยแสดงผล ที่ LCD บนตัว ST.No. 01

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

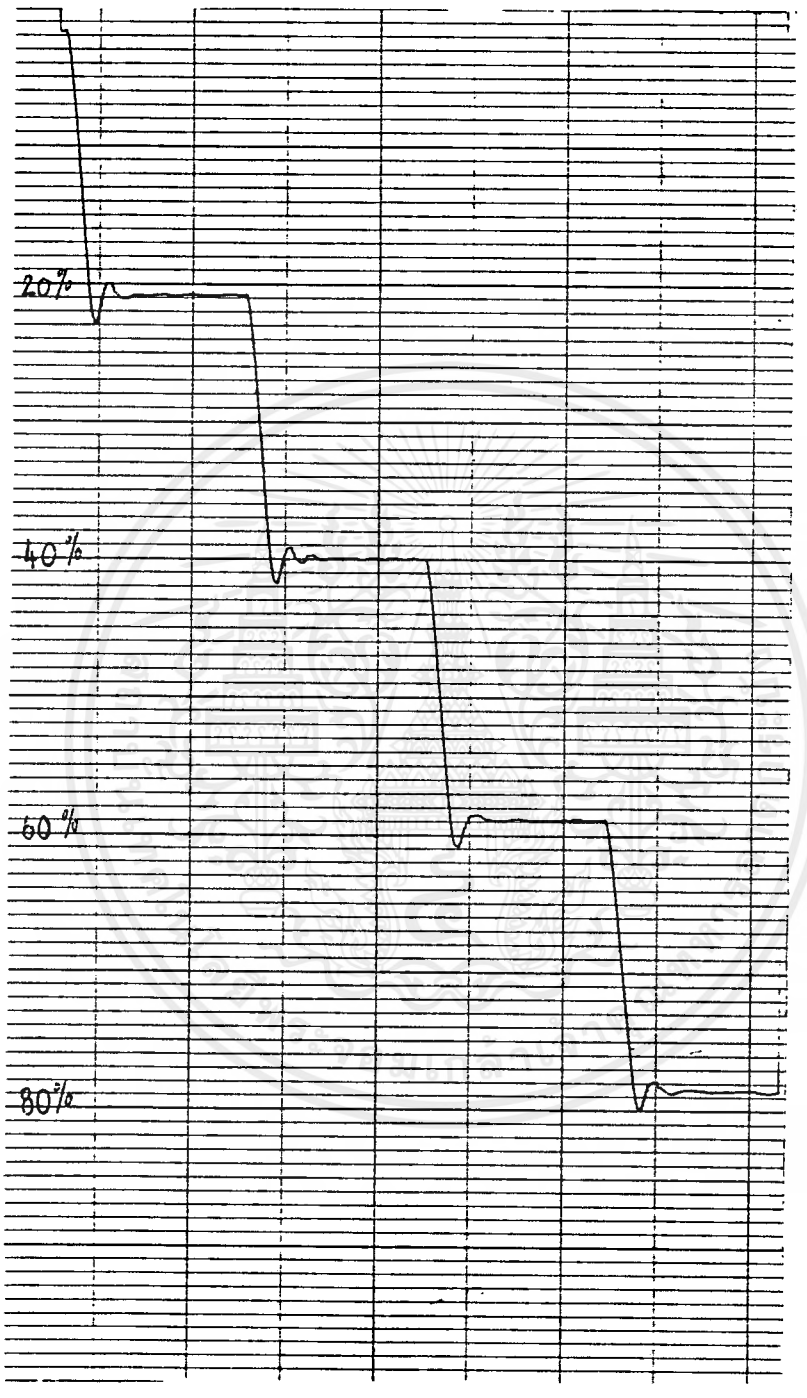
System Software



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทดลองโหมด PID โดยการควบคุมกระบวนการระดับน้ำ

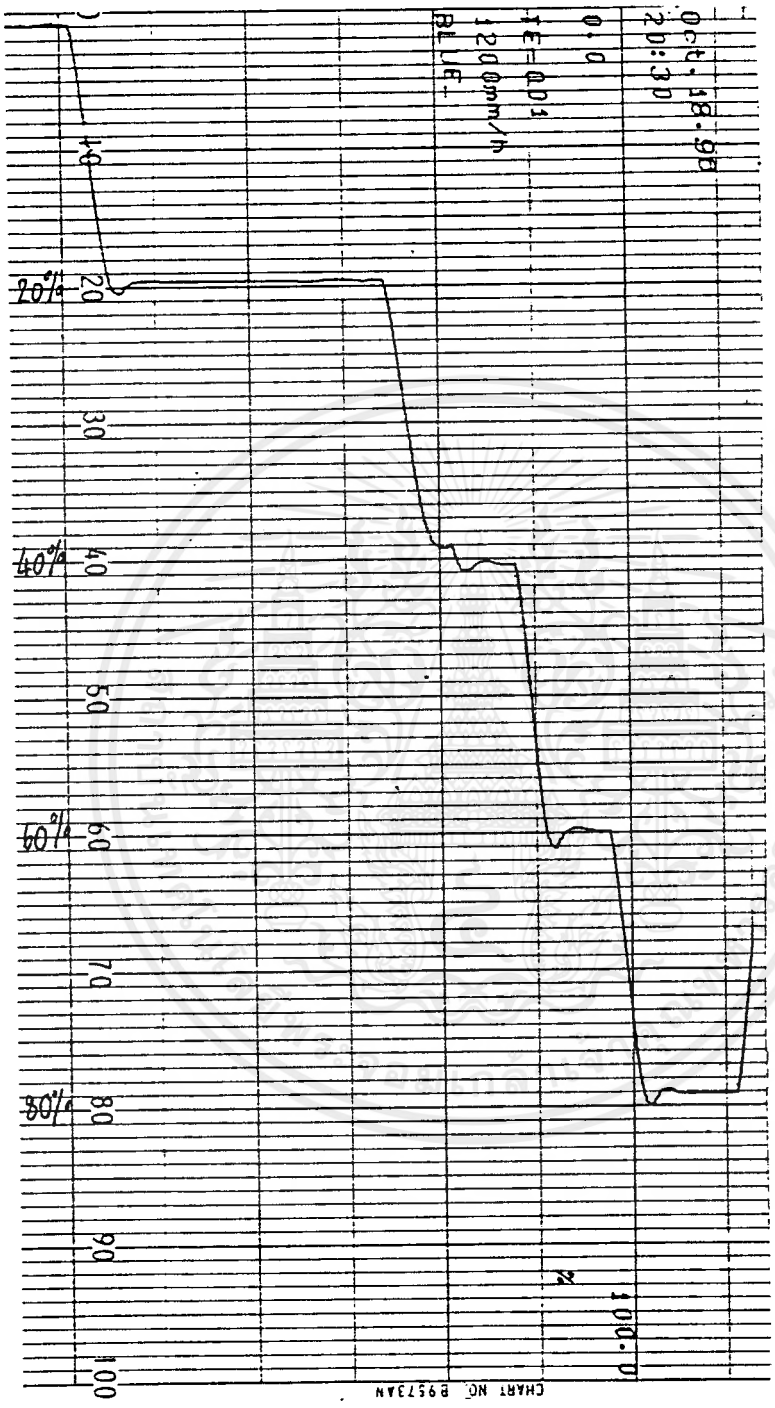
การทดลองที่ 1



Kp = 1 , Ki = 0.25 , Kd = 0.05

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทดลองที่ 2



Kp = 2 , Ki = 0.1 , Kd = 0.05

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปัญหาที่เกิดขึ้นในการทำ Analog I/O Controller (PID)

1. อุปกรณ์ในการทำวงจรต่าง ๆ ที่ซื้อจากบ้านหม้อมีคุณภาพต่ำ ไม่ได้ตามมาตรฐานที่กำหนด ทำให้ การทำงานของวงจร ไม่เที่ยงตรงเท่าที่ควร

2. เครื่องมือ และอุปกรณ์ ในการทำงานมีน้อย และบางชิ้นเสียหายบางส่วน เช่น มิเตอร์บางครั้งวัดได้ บางครั้งวัดไม่ได้ และในการตรวจสอบบางอย่างก็ไม่มีเครื่องมือที่สามารถตรวจสอบได้ เช่น Protocal การสื่อสาร ไม่สามารถใช้มิเตอร์วัดได้ทำให้เกิดตัวแปรในการหาจุดบกพร่องมากขึ้น จึงทำให้เสียเวลา และทำให้อุปกรณ์ในวงจรเสียหายมากขึ้น

3. การเลือกใช้กล่องที่เล็กเกินไปทำให้ลงอุปกรณ์ได้ยาก และมีโอกาส ลัดวงจรเสียหายได้ง่าย การไล่สายสัญญานทำได้ยาก

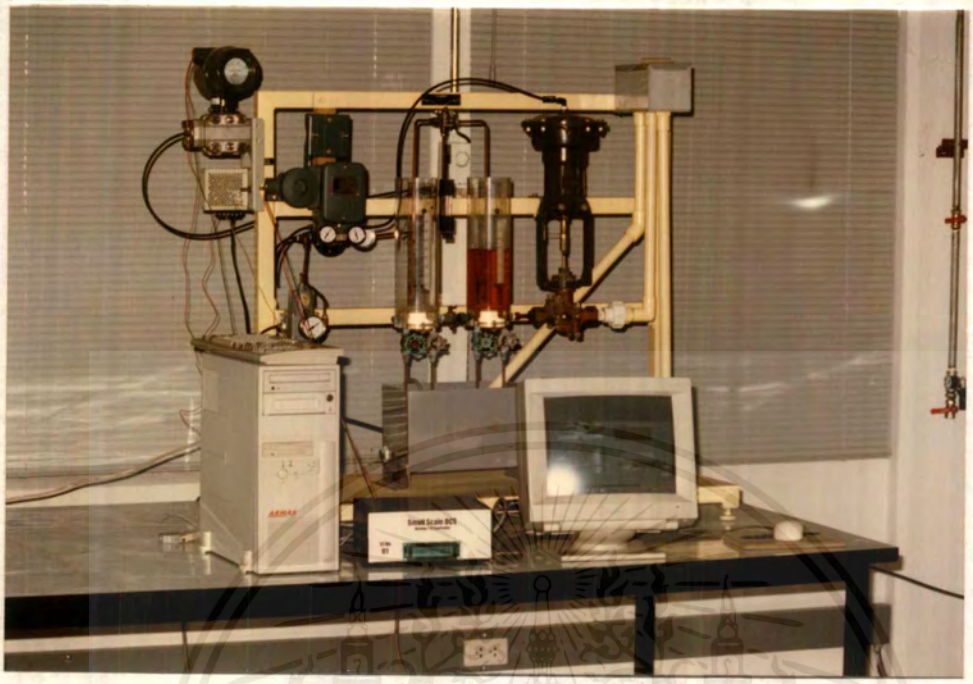
4. ความเร็ว ของ Analog I/O Controller (PID) ที่ใช้การประมวลผลจากคอมพิวเตอร์ เป็นหลักจะช้ากว่า ที่ใช้การประมวลผลจาก ไมโครโปรเซสเซอร์ ในตัว เนื่องจากเราใช้การสื่อสารข้อมูลผ่าน สายอนุกรม

5. RS-485 ทำงานไม่สมบูรณ์ หาข้อมูลและที่ปรึกษาไม่ได้

6. การเขียน โปรแกรมบางส่วนต้องรอทดสอบกับ HARD WARE เพราะฉะนั้นการล่าช้าของการทำ HARD WARE จึงมีผลให้การเขียนโปรแกรมล่าช้าไปด้วย

7. การอัดโปรแกรมลง EPROM มีขั้นตอนที่ลำบากเมื่ออัดไม่สำเร็จการล้างก็ ยุ่งยากและเสียเวลามาก

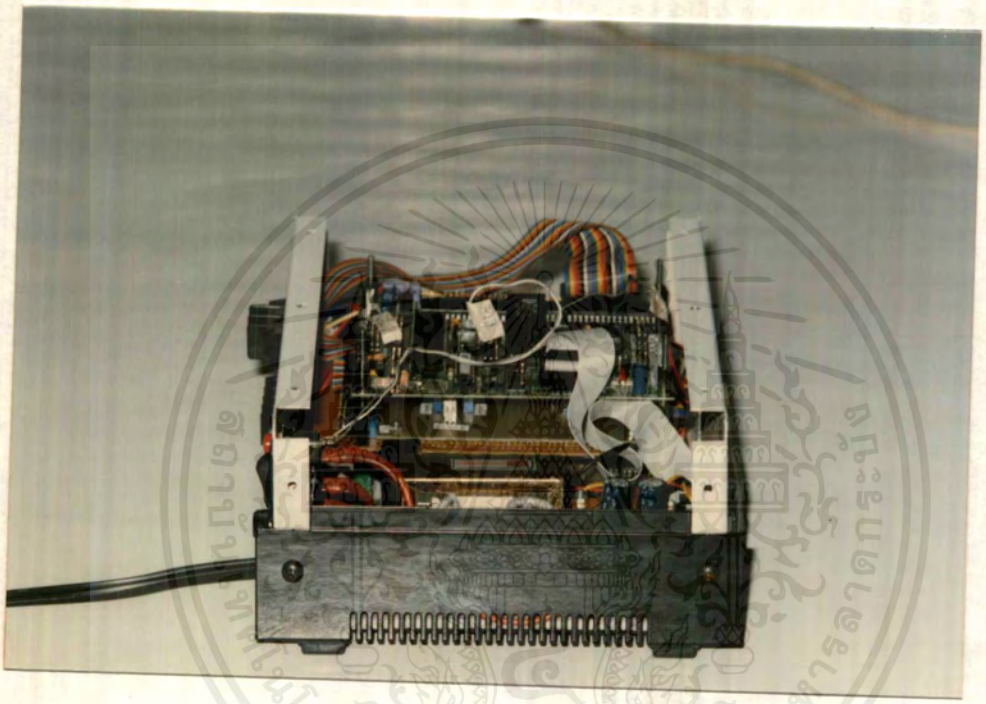
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปแสดง UNIT 01 (ANALOG I/O CONTROLLER) ทำการต่อควบคุมกระบวนการระดับน้ำ

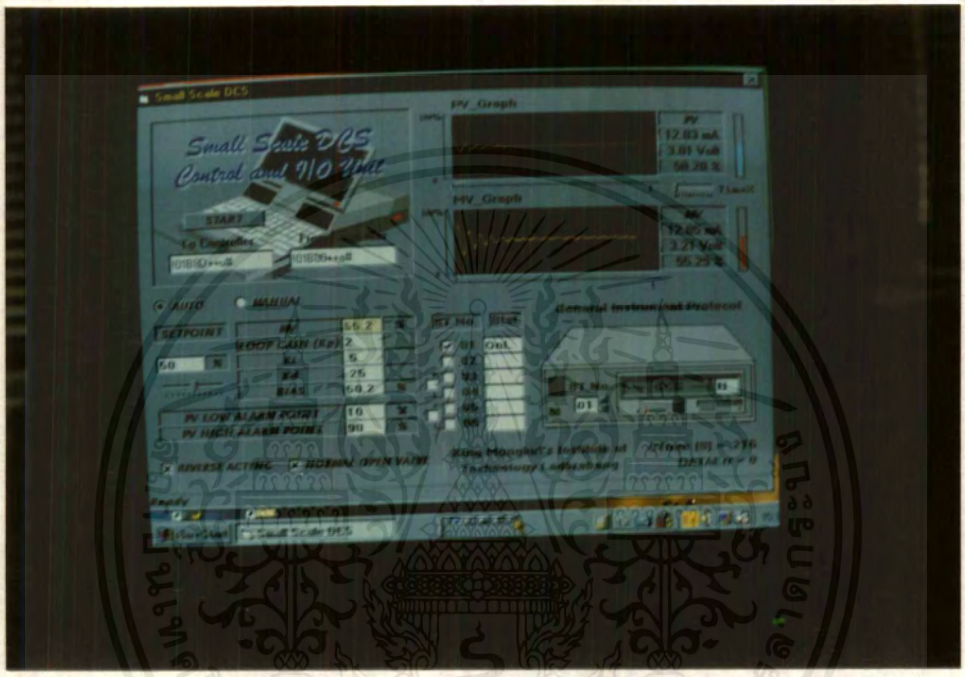
รูปแสดง UNIT 01 (ANALOG I/O CONTROLLER)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปแสดงการต่อวงจรภายในของ UNIT 01

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปแสดงหน้าจอของ ST.No. 01

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 8

Data Linker (RS-232 To RS-485 Converter)

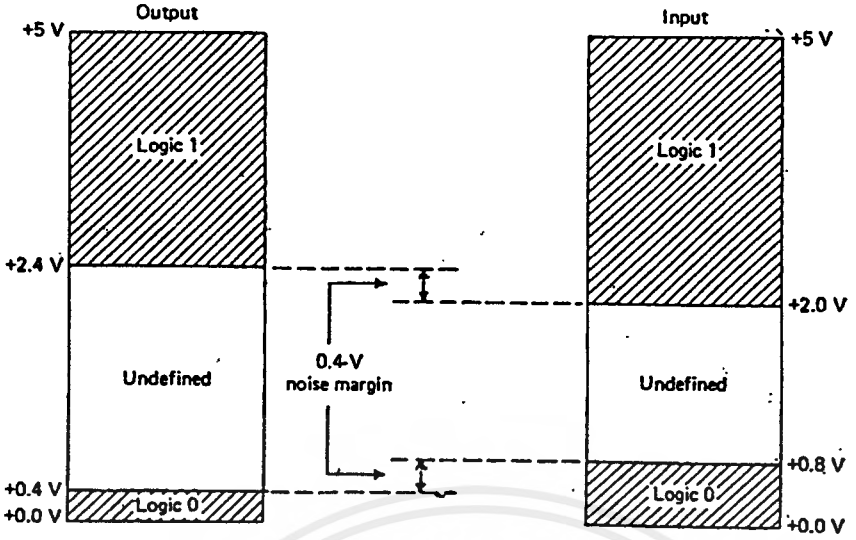
8.1 มาตรฐานการ Interface

8.1.1.ระดับสัญญาณ TTL (Transistor Transistor Logic)

โดยทั่วไปข้อมูลจะถูกแทนด้วยเลขฐานสอง ระดับแรงดัน + 5 V จะแทนด้วย ลอจิก 1 และระดับแรงดัน 0 V จะแทนด้วยลอจิก 0 การแทนลอจิกด้วย ระดับแรงดันเหล่านี้ เรียกว่า การแทนระดับสัญญาณของอุปกรณ์ TTL (Transistor- Tansistor Logic) ซึ่งจะใช้ทั่ว ๆ ไป ในการส่งข้อมูลระหว่างอุปกรณ์ชิ้นหนึ่งไปยังอุปกรณ์อีกชิ้นหนึ่งในเครื่องคอมพิวเตอร์ แต่มันจะไม่เหมาะสมที่จะกำหนดระดับแรงดัน ในการแทนลอจิก 0 หรือ 1 ที่จำเพาะเจาะจงลงไป เป็นค่าเฉพาะเพียงค่าเดียว ดังนั้นระดับแรงดันที่ใช้แทนลอจิก 0 และ 1นั่นจึงถูกกำหนดเป็นพิสัย (Range) ดังแสดงไว้ในรูปที่ 8.1 จากรูปจะเห็นว่า พิสัยของแรงดันที่ส่งจากอุปกรณ์ที่เป็นตัวส่งจะแตกต่าง จากพิสัยของแรงดันที่ถูกรับโดยอุปกรณ์ที่เป็นตัวรับ นั่นคือ ตัวส่ง (Driver)จะต้องจ่ายแรงดันที่มีระดับสัญญาณต่ำที่สุดเท่ากับ 2.4 V ในการส่งลอจิก 1 แต่ตัวรับ (Receiver) จะถือว่าระดับแรงดันที่มีระดับสัญญาณอยู่ระหว่าง 2.0 ถึง 2.4 V เป็นลอจิก 1 ด้วย สำหรับสาเหตุที่ถือว่าระดับสัญญาณในช่วง 2.0 - 2.4 V เป็นลอจิก 1 ก็เนื่องจากมีการสูญเสีย (Loss) ของสัญญาณระหว่างตัวส่งและตัวรับขึ้น ความคลาดเคลื่อนของ ระดับแรงดันที่เกิดขึ้นนี้เรียกว่า " Noise Margin " ซึ่งมีค่าเท่ากับ 0.4 V ไม่ว่าจะเป็นกรณีของลอจิก 0 หรือ 1 สำหรับลอจิก 0 ความคลาดเคลื่อนจะเกิดขึ้นหลังจากสัญญาณลบกวน (Noise) ที่ปนเข้ามาในทางปฏิบัติ ความคลาดเคลื่อนนี้ถูกยอมรับให้ใช้ได้ในการใช้งานทั่ว ๆ ไป

จากรูปที่ 8.1 จะมีระดับแรงดันอยู่ช่วงหนึ่ง เรียกว่า ช่วงการเปลี่ยนสถานะซึ่งไม่อาจระบุได้ว่าสัญญาณในช่วงนั้นเป็นลอจิก 0 หรือ 1 ช่วงของสัญญาณนี้ทางด้านรับจะมีช่วงแคบกว่าทางด้านส่ง ดังนั้นระดับแรงดันที่ใช้แทนลอจิก 0 หรือ 1 จะมีช่วงกว้าง ๆ กว่าด้านส่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 8.1 คุณสมบัติทางไฟฟ้าของการอินเตอร์เฟสที่ใช้แทนระดับสัญญาณ TTL

ในการส่งข้อมูลภายในเครื่องคอมพิวเตอร์ เราจะถือว่าระดับสัญญาณที่ใช้ส่งและรับ (ซึ่งใช้การแทนระดับสัญญาณแบบ TTL) เป็นแบบอุดมคติ เนื่องจากเหตุผลต่อไปนี้

1. กำลังงานที่ใช้และการกระจายความร้อนมีค่าต่ำ

2 สัญญาณที่ใช้เป็นระดับสัญญาณลอจิกแบบ TTL ซึ่งสามารถจ่ายให้แก่ ชิฟไอซี ได้โดยตรงโดยไม่ต้องใช้ Line Driver และวงจรรับข้อมูลที่มีราคาแพง

3. การอินเตอร์เฟสระหว่างอุปกรณ์ TTL จะทำงานที่ความถี่สูง ซึ่งจำเป็นต้องใช้ในการส่งข้อภายในคอมพิวเตอร์

ปัญหาที่เกิดขึ้นในการสื่อสารข้อมูลระหว่างอุปกรณ์แต่ละชิ้นที่ไม่ได้อยู่ในเครื่อง ๆ เดียวกัน เช่น การติดต่อเครื่องคอมพิวเตอร์ในสำนักงานเดียวกัน เทคนิคที่ใช้ในการสื่อสารข้อมูลภายในเครื่องนั้นไม่เพียงพอที่จะนำไปประยุกต์ใช้กับการติดต่อระหว่างเครื่องหรืออุปกรณ์แต่ละชิ้นได้ เราต้องเพิ่มเทคนิคบางอย่างเข้าไปอีกเพื่อให้การสื่อสารระหว่างเครื่องเป็นไปได้อย่างถูกต้องในปัจจุบันมีเทอร์มินัลบางตัวที่ใช้ระดับสัญญาณแบบ TTL กับคอมพิวเตอร์หลัก ซึ่งถ้าทำการสื่อสารข้อมูลด้วยระดับสัญญาณนี้ ในระยะทางมากกว่า 2 - 3 ฟุต อาจจะมีปัญหาหรือข้อยุ่งยากบางข้อเกิดขึ้น เนื่องจาก

1. ระดับสัญญาณ TTL มักถูกเหนี่ยวนำจากสัญญาณรบกวนภายนอกได้ง่าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. การสูญเสีย (Loss) ไปในสายทำให้ระดับของแรงดันของสัญญาณที่ส่งออกไปลดลง ซึ่งมีผลกระทบต่อระดับแรงดัน 0 - 5 V ของ TTL เพราะอาการสูญเสียระดับแรงดันไปเพียง 2-3 V สามารถทำให้ลอจิกต่าง ๆ ที่ได้รับผิดพลาดไป

8.1.2 การอินเตอร์เฟส

การส่งข้อมูลอยู่เฉพาะภายในเครื่องคอมพิวเตอร์สามารรถกระทำได้ง่ายเนื่องจากเราสามารถคาดเดาสภาพแวดล้อมภายในเครื่องได้แต่ในการส่งข้อมูลสู่ภายนอกเราไม่สามารถคาดการณ์ได้ ว่าเครื่องคอมพิวเตอร์และตัวข้อมูลจะต้องพบกับสภาพเช่นไร และมีผลกระทบต่อตัวข้อมูลและเครื่องคอมพิวเตอร์อย่างไร ดังนั้นจะต้องออกแบบวงจรเพื่อแยกข้อมูลและคอมพิวเตอร์ออกจาสภาพแวดล้อม และสัญญาณรบกวนที่ไม่ต้องการ นั่นคือตัวอินเตอร์เฟสซึ่งมีหน้าที่เป็นจุดเชื่อมต่อระหว่างคอมพิวเตอร์กับอุปกรณ์ภายนอกคล้ายกับเป็นประตูของเครื่องคอมพิวเตอร์บางตรั้งเรียกว่า I / O พอร์ต (I / O Port) หรือเรียกสั้น ๆ ว่า พอร์ต (Port)

ตัวอย่างการอินเตอร์เฟสที่เห็นได้ง่าย คือ การอินเตอร์เฟสระหว่างบ้านกับการไฟฟ้า การไฟฟ้าจะส่งไฟฟ้าแรงสูงไปตามสาย และลดสู่ระดับที่ปลอดภัยเพื่อจ่ายเข้าบ้านผู้ใช้ การไฟฟ้าจะเป็นผู้กำหนดในการต่อเข้ากับระบบไฟฟ้าดังนั้นก่อนการต่อเข้ากับระบบไฟฟ้า ผู้ใช้จะต้องตรวจสอบขนาดของสายไฟ และอุปกรณ์ป้องกันว่าตรงกับข้อกำหนดหรือไม่ ในตัวอย่างนี้ การอินเตอร์เฟสทำหน้าที่ 3 อย่างคือ เป็นการป้องกันความเสียหาย ต่อระบบการส่งจ่ายไฟของการไฟฟ้า เป็นการป้องกันอันตรายของผู้ใช้ เนื่องจากไฟฟ้าแรงสูง และเป็นตัวจัดระบบให้เกิดความเหมาะสมของการใช้งานและความปลอดภัย

การป้องกันอุปกรณ์เป็นเพียงจุดมุ่งหมายหนึ่งของการอินเตอร์เฟส วัตถุประสงค์หลักของการอินเตอร์เฟสก็คือ การใช้อุปกรณ์อินเตอร์เฟสเป็นสื่อกลางของการส่งข้อมูล และความง่ายสะดวกต่อการใช้งาน หากวงจรที่สามารถป้องกันเครื่องคอมพิวเตอร์และข้อมูล แต่การใช้งานยากวงจรอินเตอร์เฟสก็แทบไม่มีประโยชน์เลย เพราะมันไม่ได้เพิ่มความสะดวกต่อผู้ใช้

เมื่อความสามารถมทำการอินเตอร์เฟสได้สำเร็จก็สามารถส่งข้อมูลสู่ภายนอกได้ แต่ก็มีปัญหาที่เกิดขึ้นคือ การส่งข้อมูลแบบขนาน ถ้าส่งในระยะทางไกล จะเสียค่าใช้จ่ายสูง และการสูญหายของข้อมูลทำให้การใช้งานของมันถูกจำกัดอยู่กับอุปกรณ์เพียงไม่กี่ชนิด เช่น เครื่องพิมพ์ที่มักจะอยู่ใกล้กับเครื่องคอมพิวเตอร์และต้องทำงานที่ความเร็วสูง แต่ยังใช้งานส่งข้อมูลแบบขนานภายในเครื่องคอมพิวเตอร์เนื่องจากไม่ต้องใช้สายไฟขนาดยาว

ทางเลือกอีกทางหนึ่งแทนการส่งข้อมูลทุกบิตในเวลาเดียวกันทีท่ใช้สายไฟหลายเส้น คือ การส่งข้อมูลที่ละบิต และข้อมูลจะถูกต่อรวมเข้าเป็นไบต์ใหม่ การส่งข้อมูลที่ละบิต ทำให้สามารถ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใช้สายไฟเพียงสองเส้นในการส่งข้อมูล ซึ่งช่วยให้เราประหยัดค่าสายไฟได้มาก การส่งข้อมูลวิธีนี้เรียกว่า การส่งข้อมูลแบบอนุกรม การส่งข้อมูลด้วยวิธีนี้ ช่วยให้เราประหยัดค่าใช้จ่ายแต่ทำให้ประสิทธิภาพการทำงานลดลงไปด้วย ความเร็วที่ลดลงไม่ใช่ส่วนสำคัญมากนัก สำหรับการใช้งานทั่วไป หากพิจารณาถึงอุปกรณ์ทั่วไป จะพบว่าส่วนใหญ่จะทำงานช้ามาก เมื่อเทียบกับความเร็วในการทำงาน ภายในไมโครโปรเซสเซอร์ อุปกรณ์ทั่วไปมักจะเป็นกระบวนการทางกลไก (Machine) เป็นตัวจำกัดความเร็วของเครื่องลงไปอย่างมาก เช่น ความเร็วของเครื่องพิมพ์ถูกจำกัดที่ความเร็วของหัวพิมพ์ (Printhead) และความเร็วของดิสก์ไดรฟ์ถูกจำกัดโดยอัตราเร็วของการหมุนของไดรฟ์ ดังนั้นความเร็วในการส่งข้อมูลแบบขนานจะเสียไปโดยเปล่าประโยชน์ เมื่อนำมาใช้กับอุปกรณ์ประเภทนี้ ดังนั้นการส่งข้อมูลแบบอนุกรมสามารถนำมาใช้งานได้แม้ว่าอัตราเร็วการส่งข้อมูลจะลดลง แต่ก็ยังสามารถใช้งานร่วมกับอุปกรณ์ประเภทนี้ได้อย่างมีประสิทธิภาพ ข้อเสียจากความเร็วที่ลดลงไปไม่อาจจะเทียบได้กับผลที่ได้จากคุณภาพการส่งและระยะทางการส่งที่เพิ่มขึ้น

8.1.3 มาตราฐานการอินเตอร์เฟส

ในขั้นต้นกล่าวเพียงว่า เรามีความจำเป็นที่จะต้องมีการอินเตอร์เฟส ระหว่างเครื่องคอมพิวเตอร์ กับอุปกรณ์อื่น ๆ ภายนอก ควรใช้วิธีการส่งข้อมูลแบบอนุกรม เนื่องจากวิธีการออกแบบที่ "ถูกต้อง" มีด้วยกันหลายวิธีทำให้เกิดวงจรอินเตอร์เฟสขึ้นมาได้หลายแบบ ทำให้เกิดปัญหากับวงจรอินเตอร์เฟส คือความคอมแพติเบิลกับอุปกรณ์อินเตอร์เฟสอื่น ๆ

การอินเตอร์เฟสข้อมูลแบบอนุกรมมีอยู่มากมายหลายระบบตั้งแต่ในอดีตแล้ว เช่น การส่งข้อมูลข้ามทวีปที่มีมาตั้งแต่ปี 1866 ที่อาศัยสัญญาณมอร์ส (Teleprinter และ Telewriter) ทำให้เกิดอุปกรณ์การติดต่อขข้อมูลแบบอนุกรมหลายชนิด แต่ละชนิดจะมีอินเตอร์เฟสเป็นของตัวเอง เมื่อพิจารณาทางด้านอิเล็กทรอนิกส์ถึงสาเหตุของความหลากหลายของวงจรอินเตอร์เฟส สายโทรสาร (Telegraph) ที่ใช้ในการติดต่อข้อมูลเป็นตัวกลางที่สามารถต่อเข้ากับอุปกรณ์ได้หลายชนิด และสามารถทนต่อสภาพการใช้งานได้หลายรูปแบบ ต่อมาคอมพิวเตอร์ได้เข้ามามีส่วนร่วมในการสื่อสารมากขึ้น

เครื่องคอมพิวเตอร์ในยุคแรก ๆ ไม่ได้ทำงานในลักษณะนี้ I / O ส่วนใหญ่ทำโดยอาศัยบัตรเจาะรูหรือเทปกระดาษ แต่เมื่อคอมพิวเตอร์กับระบบคำเนินการเริ่มมีประสิทธิภาพมากยิ่งขึ้น ทำให้สามารถติดต่อกับมนุษย์โดยตรงต่อมาเมท่อมีระบบจัดแบ่งเวลา (Time-Sharing-System) ขึ้นมาทำให้ผู้ใช้หลายคนสามารถต่อเข้าใช้คอมพิวเตอร์เครื่องเดียวได้ในเวลาเดียวกัน การติดต่อกับคอมพิวเตอร์ในระยะที่ไม่ห่างกันมากนักจึงเป็นที่นิยมกันมากขึ้น แต่เมื่อผู้ใช้มีความต้องการที่จะติด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ต่อกับเครื่องในตะยะที่ไกลมากขึ้นจึงหันมาใช้การส่งข้อมูลผ่านสายโทรศัพท์ทำให้ผู้ใช้สามารถติดต่อกันได้แทบทุกจุด

8.1.4 มาตรฐานการอินเตอร์เฟส RS - 232

ในปี 1969 EIA (Electronic Industries Assuciation) ห้องวิจัย Bell และบรรดาผู้ผลิตอุปกรณ์สื่อสารได้ร่วมกันจัดตั้งมาตรฐาน EIA RS - 232 ต่อมาได้มีการปรับปรุงเป็น RS - 232 - C และไม่นานมานี้ได้ออกมาตรฐาน RS - 232 - D และยังมีมาตรฐานคล้ายกัน ซึ่งออกโดยองค์กรระหว่างประเทศ คือ Consultative Committee on international Telegraphy and Telephony (CCITT) เพื่อให้เข้าใจกับการอินเตอรค์เฟส RS - 232 ได้ดียิ่งขึ้น ควรที่จะเข้าใจถึงวัตถุประสงค์หลักของ RS - 232 ก่อนคือ การอินเตอร์เฟสระหว่างเทอร์มินัล (Data Terminal Equipment หรือ DTE) กับโมเด็ม (Data Communications Equipment หรือDCE) เพื่อใช้ส่งข้อมูลแบบอนุกรม ประกอบด้วย 4 ส่วนต่าง ๆ ดังนี้

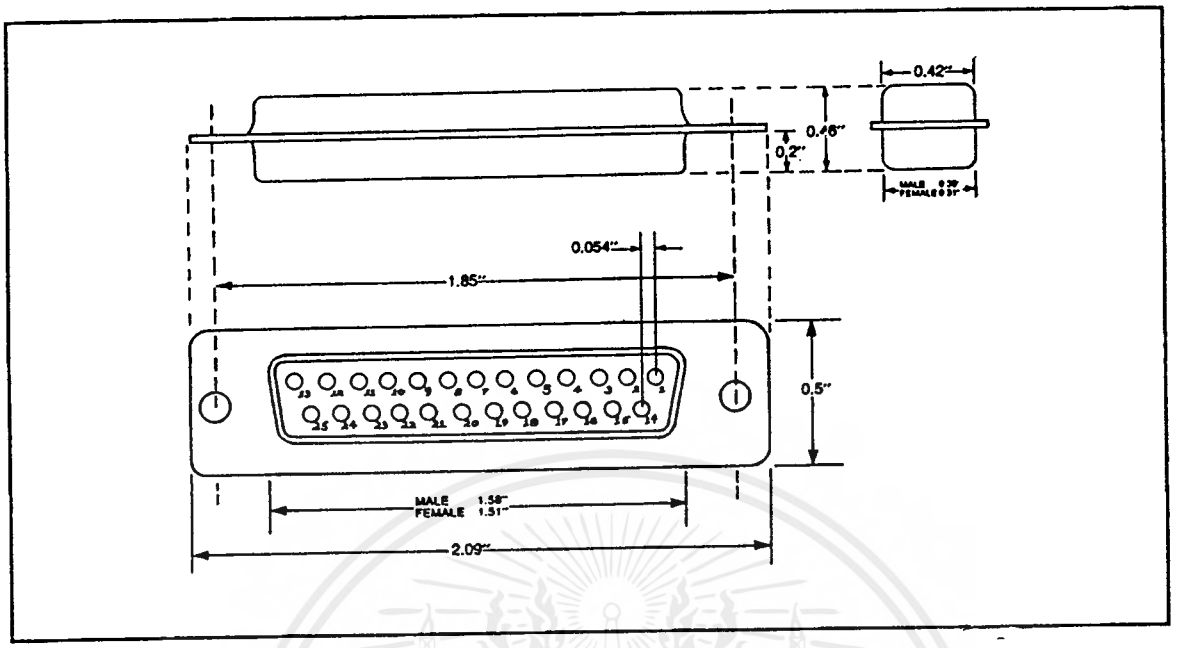
- คุณสมบัติทางไฟฟ้าของสัญญาณ (Electrical Signal Characteristics)

ในส่วนนี้จะอธิบายถึงรูปแบบของสัญญาณไฟฟ้าที่ตัวอินเตอร์เฟสจะส่งออกและรับเข้ามาจากภายนอกระดับแรงดันไฟฟ้าที่ใช้แทนลอจิกต่าง ๆ (0 หรือ 1 , ON / OFF , MARK / SPACE รวมถึงอัตราเร็วในการรับส่งข้อมูลและคุณสมบัติของวงจรรับและส่ง

- คุณสมบัติทางกลไกการอินเตอร์เฟส (Interface Mechanical Characteristics)

ตัวอินเตอร์เฟสประกอบด้ายส่วนที่เป็นปลั๊ก (Plug) และตัวเสียบ (Receptacle) โดยเต้าเสียบจะอยู่บน DCE ใน RS - 232 - D ได้เพิ่มข้อกำหนด คอนเน็กเตอร์ DB - 25 แสดงไว้ในรูปที่ 8.2 และคุณสมบัติของสายเคเบิล เช่น ความยาวและจำนวนลวดตัวนำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 8.2 รูปภาพคอนเน็คเตอร์ DB - 25

- หน้าที่การทำงานของวงจรการแลกเปลี่ยน (Functional Discription of Interchange Circuit)

ในส่วนนี้กำหนดหน้าที่และตั้งชื่อให้กับสัญญาณไฟฟ้าต่าง ๆ ที่นำมาใช้ เช่น Transmitted Data (ข้อมูลส่งออก) ได้ถูกกำหนดไว้ที่ ขา 2 ซึ่งข้อกำหนดมีมากถึง 25 ข้อ แต่มีเพียงไม่กี่ข้อที่เกี่ยวข้องกับไมโครคอมพิวเตอร์

- มาตรฐานการอินเตอร์เฟสสำหรับระบบการสื่อสารเฉพาะอย่าง (Standard Tnterfaces For Selected Communications System Configurations)

ในส่วนนี้เป็นรายละเอียดต่าง ๆ สำหรับการติดต่อระหว่างโมเด็มกับเทอร์มินัลทั่วไป

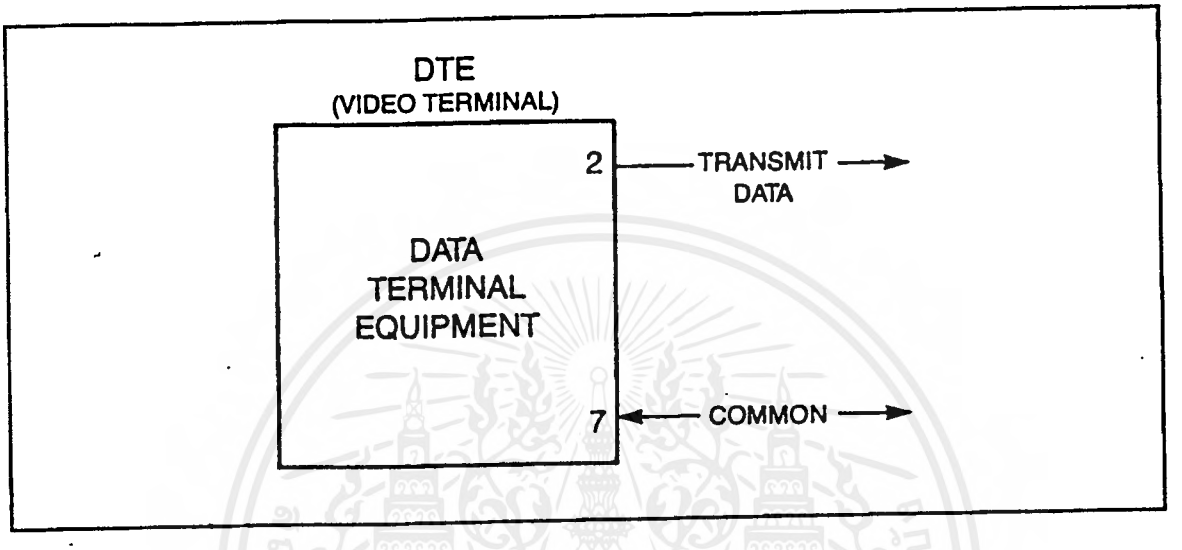
8.1.5 พื้นฐานการอินเตอร์เฟส

การอินเตอร์เฟส RS - 232 จะประกอบด้วยสายไฟ 2 เส้น เส้นหนึ่งเป็นเส้นสำหรับรับส่งข้อมูลและอีกเส้นหนึ่งเป็นเส้นสำหรับอ้างอิงระดับแรงดันของวงจรอินเตอร์เฟส (Circuit Common) ในส่วนแรงดันอ้างอิงนี้มักมีผู้ไขว้เขวว่าเป็นสิ่งเดียวกับกราวด์ (Ground) ขาอ้างอิง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แรงดัน (ขา 7) ในวงจรอินเตอร์เฟสเป็นสิ่งที่ขาดไม่ได้ไม่ว่าวงจรนั้นจะซับซ้อนหรือง่ายดายเพียงใด

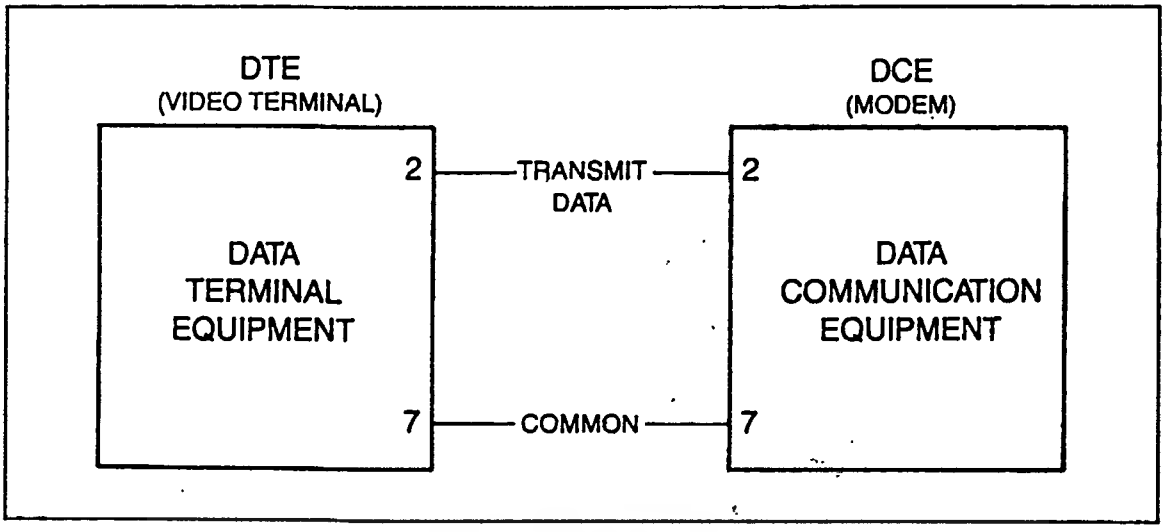
รูปที่ 8.3 แสดงอุปกรณ์ DTE เบื้องต้น (Data Terminal Equipment) ซึ่งเป็นตัวเทอร์มินัลที่ประกอบไปด้วยคีย์บอร์ด และจอมอนิเตอร์ ตัวเลขแสดงภายในอุปกรณ์จะแสดงหมายเลขขาของตัวคอนเน็กเตอร์ของอุปกรณ์ " TRANSMIT DATA " หมายถึง การส่งสัญญาณออกจากตัวส่ง (Transmitter) ไปยังตัวรับสัญญาณ (Receiver)



รูปที่ 8.3 แสดงอุปกรณ์ DTE เบื้องต้น

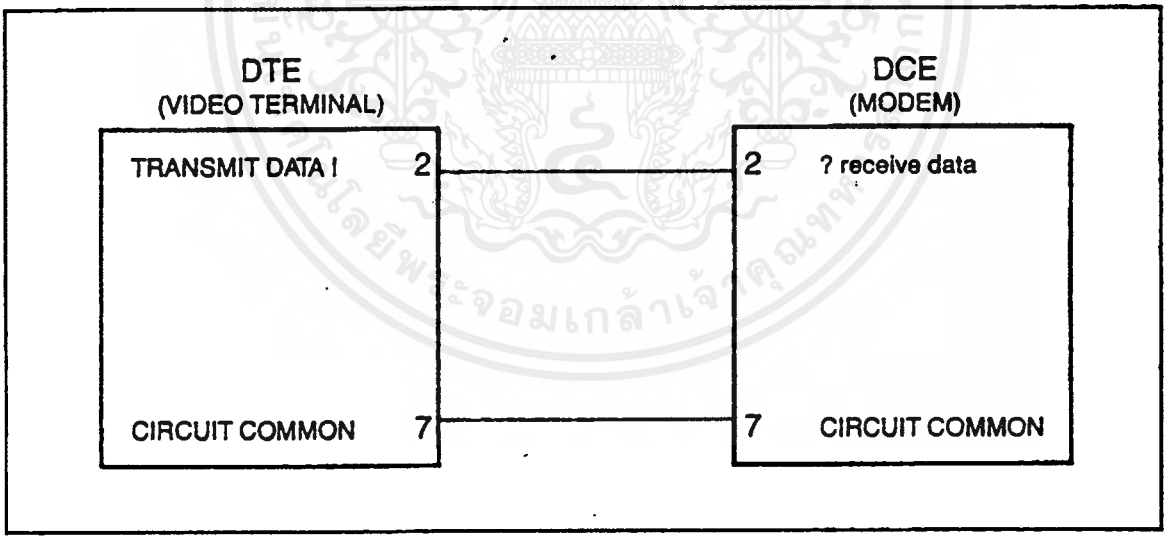
อินเตอร์เฟสใด ๆ ก็ตามจะประกอบด้วยส่วนของตัวส่งข้อมูลและส่วนของตัวรับข้อมูล รูปที่ 8.4 สมมุติให้ส่วนรับข้อมูลเป็นอุปกรณ์ DCE (Data Communication Equipment) ดั้งเดิม ซึ่งก็คือโมเด็มนั่นเอง เมื่อพิจารณาตามรูป พบว่าข้อมูลที่ส่งออกจากขา 2 ของอุปกรณ์ DTE จะรับเข้าไปยังอุปกรณ์ DCE ทางขา 2 เช่นกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 8.4 แสดงส่วนของตัวรับและตัวส่งข้อมูล

รูปที่ 8.5 ข้อมูลที่ส่งออกจากขา 2 ของ DTE จะเป็นข้อมูลตัวเดียวกับข้อมูลที่รับเข้ามาทางขา 2 ของDCE ต่อไปนี้สัญญาณที่ส่งออกไปจะเรียกว่า " เอาท์พุท " จะถูกแทนด้วยอักษรตัวพิมพ์ใหญ่กำกับด้วยเครื่องหมายอัศเจรีย์ (!) และสัญญาณที่รับเข้ามาเรียกว่า " อินพุท " จะถูกแทนด้วยอักษรตัวพิมพ์เล็กกำกับด้วยเครื่องหมายคำถาม (?)

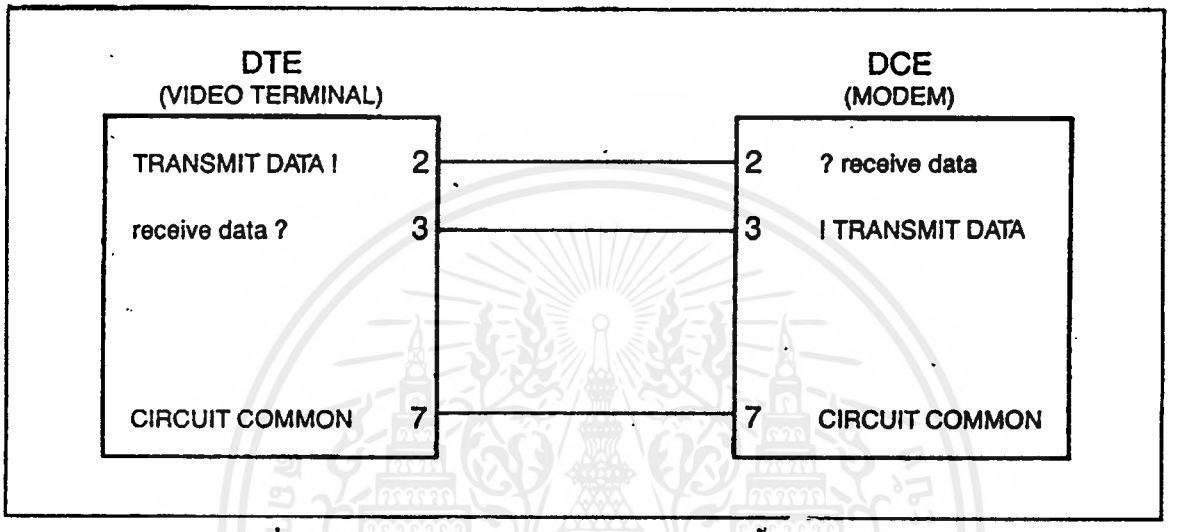


รูปที่ 8.5 อุปกรณ์ DTE และอุปกรณ์ DCE เป็นคู่อุปกรณ์ร่วมที่ทำงานตรงข้ามกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8.1.6 การรับส่งข้อมูลสองทิศทาง

รูปที่ 8.6 เมื่อเทอร์มินัลรับข้อมูลจากคีย์บอร์ดแล้วส่งต่อยังโมเด็ม โมเด็มจะส่งข้อมูลออกไปทางสายโทรศัพท์จะเห็นได้ว่าทั้งเทอร์มินัลและโมเด็มสามารถเป็นได้ทั้งอุปกรณ์รับและส่งข้อมูลและยังสามารถรับและส่งข้อมูลในทิศทางตรงกันข้ามได้อีกด้วย เช่น เมื่อโมเด็มรับข้อมูลจากสายโทรศัพท์ก็จะทำการส่งต่อไปให้เทอร์มินัล เมื่อเทอร์มินัลได้รับเอาทพุทของโมเด็มก็จะทำการแสดงผลขึ้นที่หน้าจอเทอร์มิเตอร์



รูปที่ 8.6 อุปกรณ์สามารถรับและส่งข้อมูลได้ทั้ง 2 ทิศทาง

ความแตกต่างระหว่างอุปกรณ์ DTE และ DCE ที่เราเห็นได้เบื้องต้นดังนี้

DTE ส่งเอาท์พุททางขา 2 และรับเอาท์พุททางขา 3

DCE ส่งเอาท์พุททางขา 3 และรับเอาท์พุททางขา 2

ปัญหาในการอินเตอร์เฟสไม่ว่าจะง่ายดายหรือซับซ้อนเพียงใด เราจำเป็นต้องเริ่มต้นด้วยการตรวจสอบทิศทางของสัญญาณข้อมูลที่ขา 2 และ 3 เป็นอันดับแรก (ถ้าจำกัดความโดย DTE หมายถึง อุปกรณ์ที่ข้อมูลมาสิ้นสุดและอุปกรณ์ DCE เป็นอุปกรณ์ที่ใช้ส่งผ่านข้อมูล ดังนั้นคอมพิวเตอร์ซึ่งสามารถทำได้ทั้งรับ และส่งข้อมูล จึงไม่อาจระบุได้แน่ชัดว่า เป็นอุปกรณ์ DCE หรือ DTE)

8.1.7 การแฮนด์เชค

การอินเตอร์เฟสยังมีเรื่องที่ซับซ้อนที่ยังต้องศึกษาอีกมากสิ่งแรกที่เราขาดไม่ได้คือ วิธีการควบคุมการทำงานของอุปกรณ์ 2 ตัว ให้สัมพันธ์กันในการรับส่งข้อมูลในช่วงเวลาและ ใช้

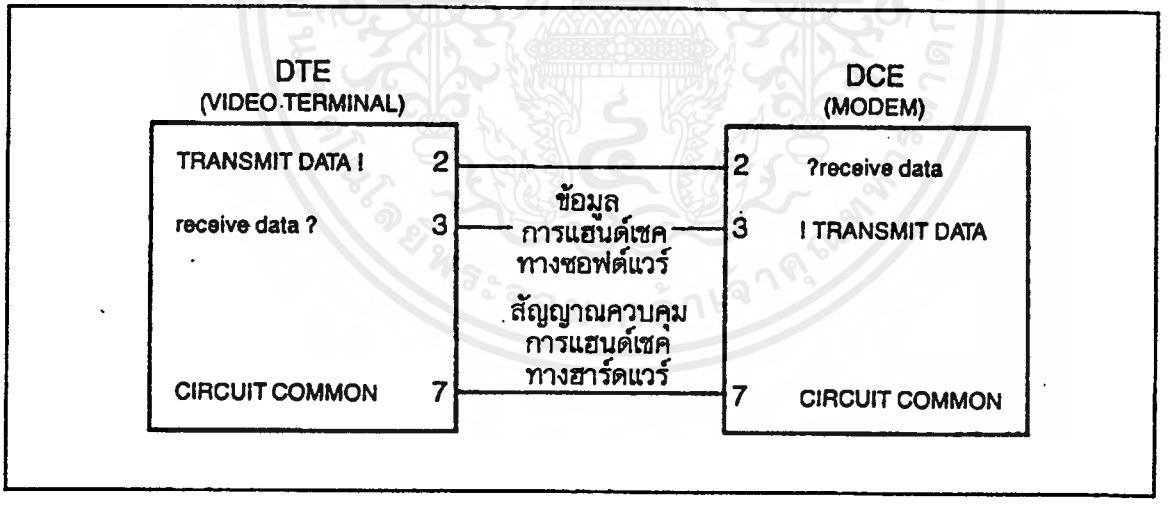
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สัญญาณที่เหมาะสม ซึ่งก็คือ กระบวนการแฮนด์เชค (Handshaking) นั่นเอง การแฮนด์เชค แบ่งได้ 2 ประเภทคือ

การแฮนด์เชคทางซอฟแวร์ (Software Handshaking) เป็นวิธีการหนึ่งในการควบคุมการทำงานของอุปกรณ์รับข้อมูลโดยส่งผ่านสัญญาณควบคุม ไปพร้อมกับตัวข้อมูลที่ต้องการส่ง เช่น ในการพิมพ์ข้อความทางเครื่องพิมพ์ คอมพิวเตอร์จะส่งข้อมูลไปที่ละบรรทัด มันจะแทรกตัวอักษรควบคุมแสดงว่าสิ้นสุดบรรทัด (End - of - Line) เพื่อแจ้งเครื่องพิมพ์ว่าขณะนี้คอมพิวเตอร์กำลังรอสัญญาณตอบกลับ ก่อนจะส่งข้อมูลบรรทัดถัดไป และเมื่อเครื่องพิมพ์รับข้อมูลเข้ามาและพิมพ์ไปจนจบบรรทัด ก็จะส่งสัญญาณไปบอกคอมพิวเตอร์ ว่ามันพร้อมจะรับข้อมความบรรทัดใหม่แล้ว

การแฮนด์เชคทางฮาร์ดแวร์ (Handware Handshakings) มีข้อจำกัดคือ จำเป็นต้องมีสายสัญญาณควบคุมต่างหาก สำหรับงานนี้โดยเฉพาะ

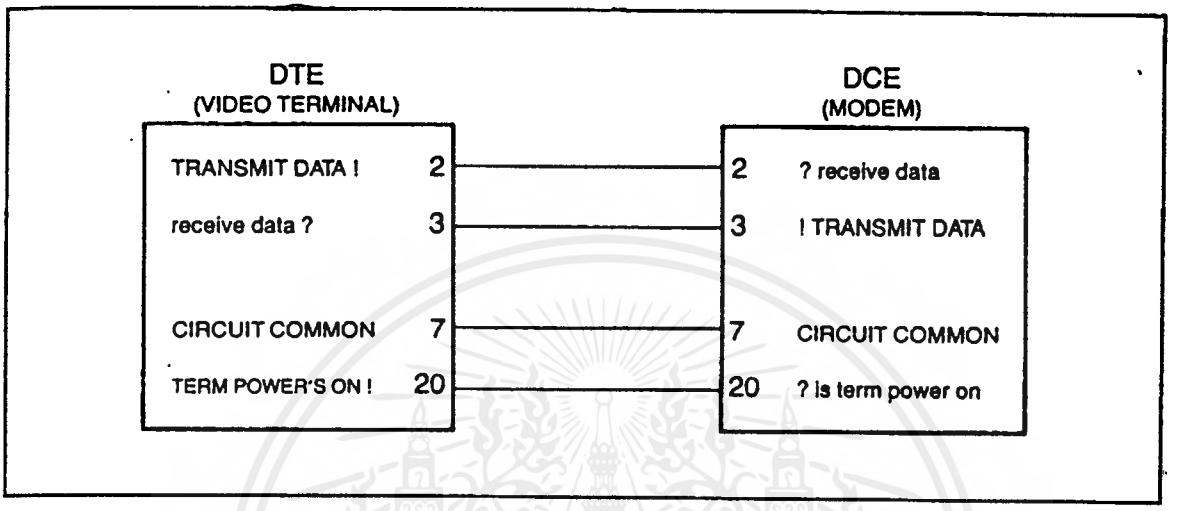
รูปที่ 8.7 สัญญาณอินเตอร์เฟสถูกแบ่งเป็น 2 ประเภทคือ สัญญาณข้อมูลคือ สัญญาณที่เป็นอักษรข้อมูลหรือข้อความที่ต้องการรับส่งจริง และสัญญาณควบคุม คือ สัญญาณอื่น ๆ ที่เหลือทั้งหมด



รูปที่ 8.7 สัญญาณควบคุมไม่มีข้อมูลเข้ามาเกี่ยวข้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เพื่อให้เกิดความเข้าใจในเรื่องการแฮนด์เชคมากขึ้น เราสมมุติให้โมเด็ม ดังภาพ 8.7 ทำงานแบบตอบโทรศัพท์อัตโนมัติ เมื่อมีโทรศัพท์เข้ามาในขณะที่โมเด็มเปิดเครื่องอยู่ แทนที่คนโทรเข้ามาจะได้รับคำว่าสวัสดี แต่กลับได้รับสัญญาณตอบรับของโมเด็มแทน ดังนั้นจึงจำเป็นต้องหาวิธีป้องกันเหตุการณ์นี้ โดยให้โมเด็มตอบโทรศัพท์ เฉพาะเมื่อคุณใช้คอมพิวเตอร์ และคอมพิวเตอร์พร้อมที่จะรับส่งข้อมูลเท่านั้น ซึ่งโมเด็มจะต้องสามารถรับรู้สถานะของคอมพิวเตอร์และตอบสนองตามแต่สถานะนั้น ๆ ได้ นั่นคือ ต้องมีการแฮนด์เชคระหว่างกัน ดังภาพที่ 8.8

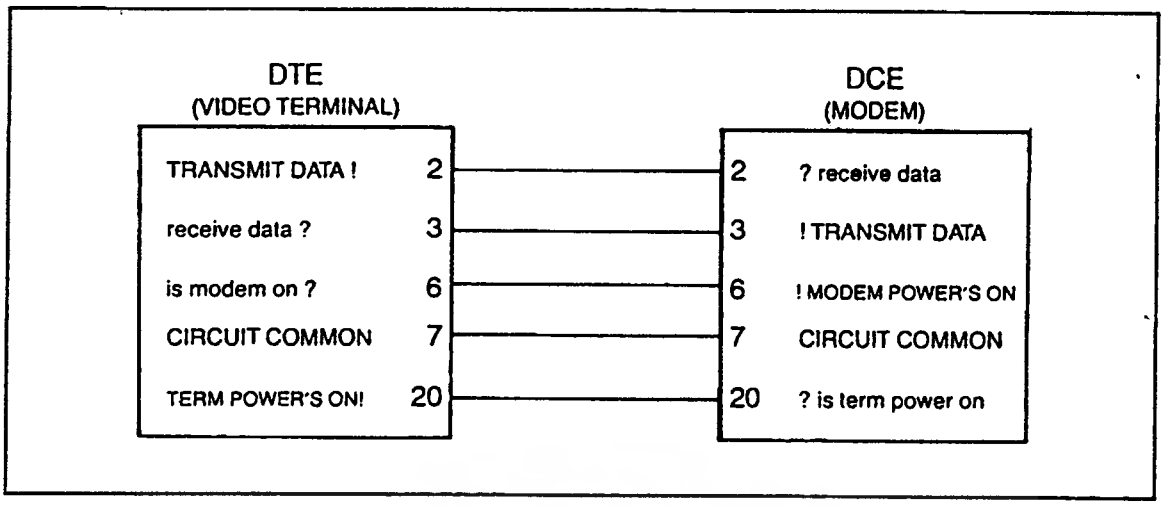


รูปที่ 8.8 โมเด็มจะตอบโทรศัพท์เฉพาะเมื่อเราเปิดเครื่องคอมพิวเตอร์แล้ว

จากรูปที่ 8.8 เมื่อเราปิดไฟเลี้ยงเทอร์มินัลแรงดันไฟที่ขา 20 จะเป็น 0 V ระดับแรงดัน 0 V จะไปปรากฏที่ขา 20 ของโมเด็มเช่นกัน เมื่อโมเด็มรับรู้ระดับสัญญาณนี้ก็จะหยุดตอบรับโทรศัพท์อัตโนมัติทันที แต่เมื่อเราเปิดสวิตช์เทอร์มินัลใหม่ วงจรภายในเทอร์มินัลจะทำให้แรงดันที่ขา 20 มีค่าระดับหนึ่ง เมื่อโมเด็มรับรู้ระดับแรงดันใหม่ก็จะกลับมาทำงานตอบโทรศัพท์อัตโนมัติอีกครั้ง

รูปที่ 8.9 สัญญาณการแฮนด์เชคที่ขาคอนแนกเตอร์ ขา 6 ทำหน้าที่คล้ายกับสัญญาณที่ขา 20 ต่างกันที่ขา 6 เอาท์พุตส่งออกมาจาก DCE และส่งไปยังอินพุทของ DTE ซึ่งการแฮนด์เชคในลักษณะนี้จะเตือนให้ผู้ใช้หากว่าลืมปิด โมเด็มก่อนใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



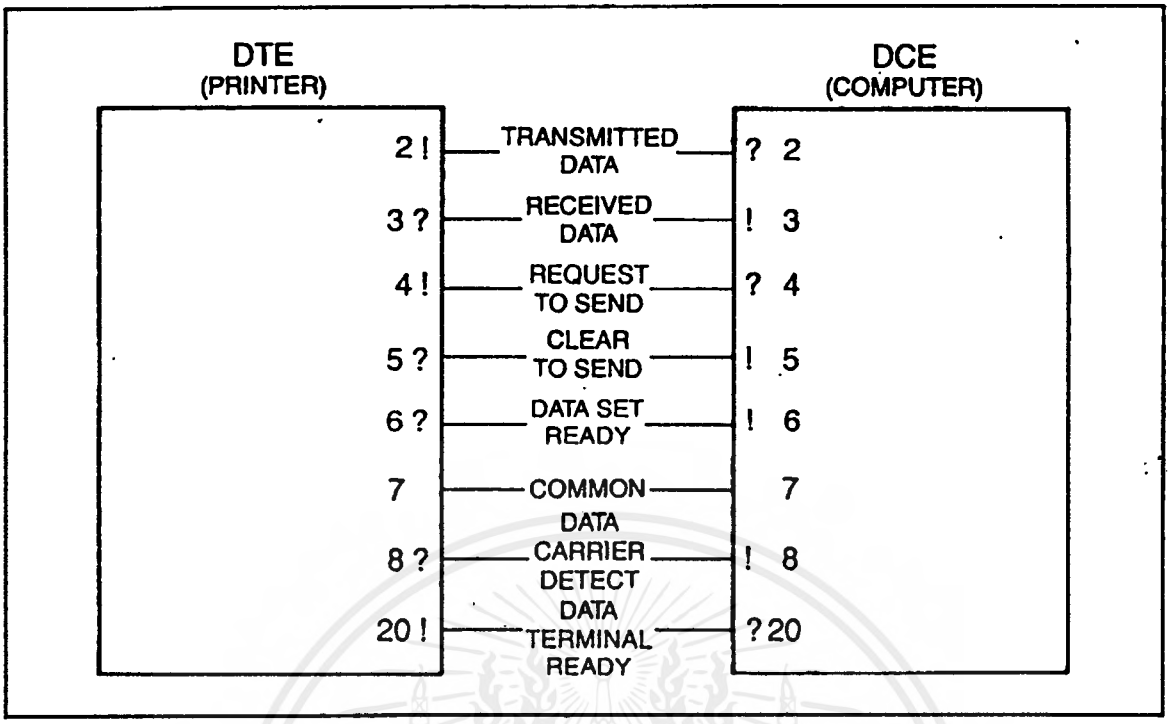
รูปที่ 8.9 คู่อินพุท / เอาท์พุตควบคุมสำหรับการแฮนด์เชค

8.2 เทคนิคที่ใช้ในการออกแบบ

8.2.1 ปัญหาในการออกแบบ

เพื่อให้เกิดความเข้าใจเกี่ยวกับการอินเตอร์เฟสระหว่าง คอมพิวเตอร์กับเครื่องพิมพ์ต้องศึกษาสัญญาณ " REQUEST TO SEND " (ขออนุญาติเปิดข้อมูล) และสัญญาณ " CLEAR TO SEND " (พร้อมรับข้อมูล) ว่ามีการทำงานอย่างไรจากภาพที่ 8.10 ในการอินเตอร์เฟสจะสังเกตเห็นว่า คอมพิวเตอร์ถูกกำหนดให้เป็นอุปกรณ์ DCE และเครื่องพิมพ์เป็นอุปกรณ์ DTE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 8.10 การอินเตอร์เฟสระหว่างเครื่องพิมพ์และคอมพิวเตอร์

สัญญาณ REQUEST TO SEND (RTS) เป็นสัญญาณ ที่ DTE ส่งไปตาม DCE ว่ามันจะส่งข้อมูลออกไปได้หรือยังเมื่อหาอินพุท Request to send (rts) ของด้าน DCE ได้รับสัญญาณขออนุญาติเข้ามาและถ้าตัว DCE พร้อมที่จะรับข้อมูล DCE จะส่งคำตอบว่าพร้อมรับข้อมูลโดยส่งสัญญาณ CLEAR TO SEND (CTS) กลับไปเมื่ออุปกรณ์ DTE พบว่าอินพุท CTS ของมันมีสัญญาณตอบรับเข้ามา มันก็จะเริ่มส่งข้อมูลให้ DCE ดังนั้น การใช้สัญญาณ RTS / CTS จึงเป็นกระบวนการทางฮาร์ดแวร์ก่อนการส่งสัญญาณจาก DTE ไป DCE

ตามมาตราฐานการอินเตอร์เฟต RS - 232 การอนิเตอร์เฟสจำเป็นต้องมีคู่สัญญาณ อินพุท / เอาท์พุทที่อุปกรณ์ DCE สามารถใช้ขออนุญาติและรอรับคำตอบในการส่งข้อมูลไปยังอุปกรณ์ DTE แต่เครื่องพิมพ์ (DTE) ไม่ค่อยจะส่งสัญญาณตอบว่าพร้อมรับข้อมูลให้กับคอมพิวเตอร์ (DCE) แต่มันกลับบอกคอมพิวเตอร์ให้ระงับการส่งข้อมูลมากกว่า

จากการที่กำหนดให้เครื่องพิมพ์เป็นอุปกรณ์ DTE และให้คอมพิวเตอร์เป็นอุปกรณ์ DCE ในการอินเตอร์เฟสทำให้ต้องมีการพบิกแพลงการอินเตอร์เฟสไปจากมาตราฐาน RS - 232

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เพื่อให้สามารถใช้งานเครื่องพิมพ์ได้ เริ่มต้นด้วยแนวความคิดที่จะกำหนดให้ขาของคอนเน็กเตอร์มีหน้าที่เฉพาะในการแฮนด์เชค ขา RTS ดูจะเหมาะสมที่สุด

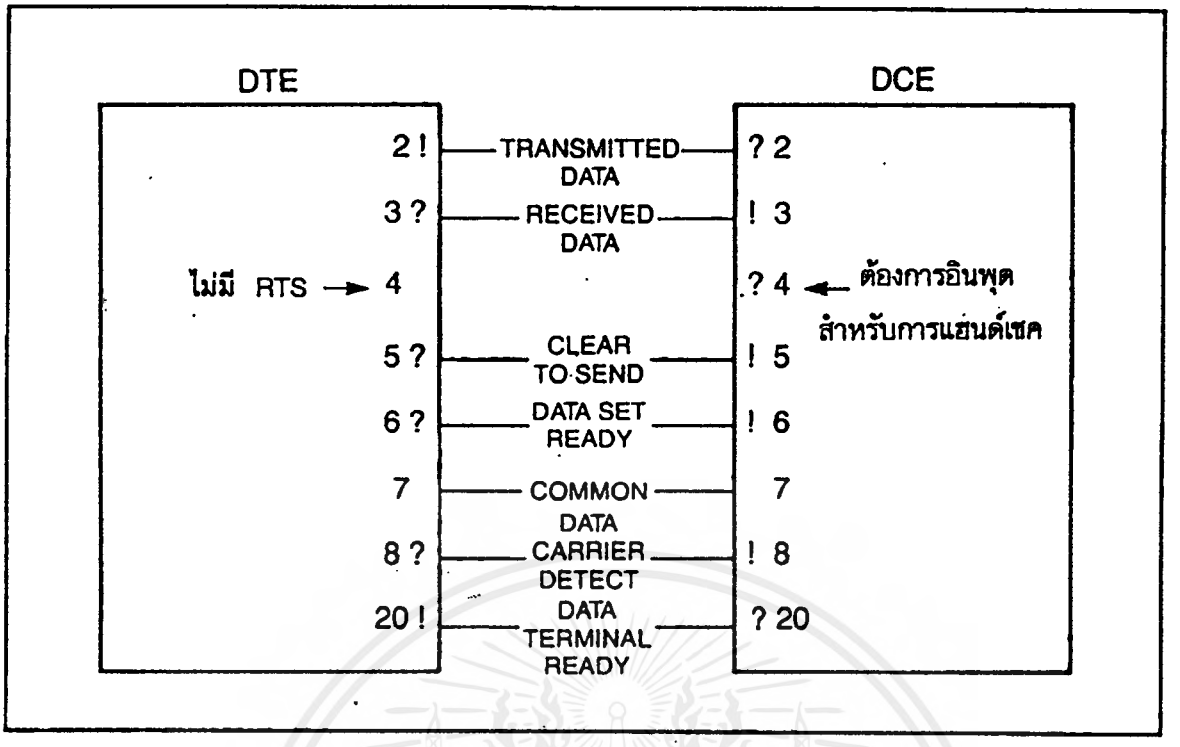
อีกประการหนึ่งที่สำคัญ คือ เมื่อจำเป็นต้องพลิกแพลงไปจาก มาตราฐานควรจะดัดแปลงในระดับใฝดจึงจะเป็นที่ยอมรับและมีเทคนิคอื่นใดที่เหมาะสมกว่าหรือไม่ เช่น ถ้าไม่ต้องการใช้การแฮนด์เชคทางซอฟต์แวร์ก็สามารถตัดสายสัญญาณ TRANSMTTTED DATA ออกไปได้ หรือ ตัดออแบบให้สัญญาณการแฮนด์เชคผ่านคอนเน็กเตอร์ขาสัญญาณ DTR ขา 20 แล้ว ก็ไม่จำเป็นต้องใช้ขา CLEAR TO SEND อีก

8.2.2 เทคนิคในการอินเตอร์เฟส

ขาของคอนเน็กเตอร์และ สัญญาณในการอินเตอร์เฟสมีอยู่เป็นจำนวนมาก แล้วแต่การเลือกใช้ต่าง ๆ กันไป เช่น บ้างก็ใช้สัญญาณควบคุม DTR / DSR หรือบ้างก็ใช้สัญญาณควบคุม RTS / CTS แทน แต่สิ่งเหล่านี้อาจทำให้เกิดปัญหาในการอินเตอร์เฟส คือ มีอินพุทด้านหนึ่งแต่ไม่มีเอาท์พุทของอีกด้านหนึ่งเข้ามา เช่น เครื่องพิมพ์อาจไม่ทำงานอาจจะไม่ทำงานหสักไม่มีสัญญาณอินพุท DSR มากระตุ้น แต่คอมพิวเตอร์ที่จะต้องต่อด้วยกลับไม่มีสัญญาณนี้ให้ในการอินเตอร์เฟส

ในกรณีอุปกรณ์ ให้แต่เอาท์พุท DTR แต่ไม่ให้ RTS ดังนั้น เมื่อต้องการต่ออุปกรณ์ DTE เข้ากับอุปกรณ์ DCE ที่ต้องภารอินพุท RTS ในการทำงาน ทำให้ไม่สามารถนำทั้งคู่ต่อเข้าด้วยกันแบบธรรมดาได้ ดังรูปที่ 8.11

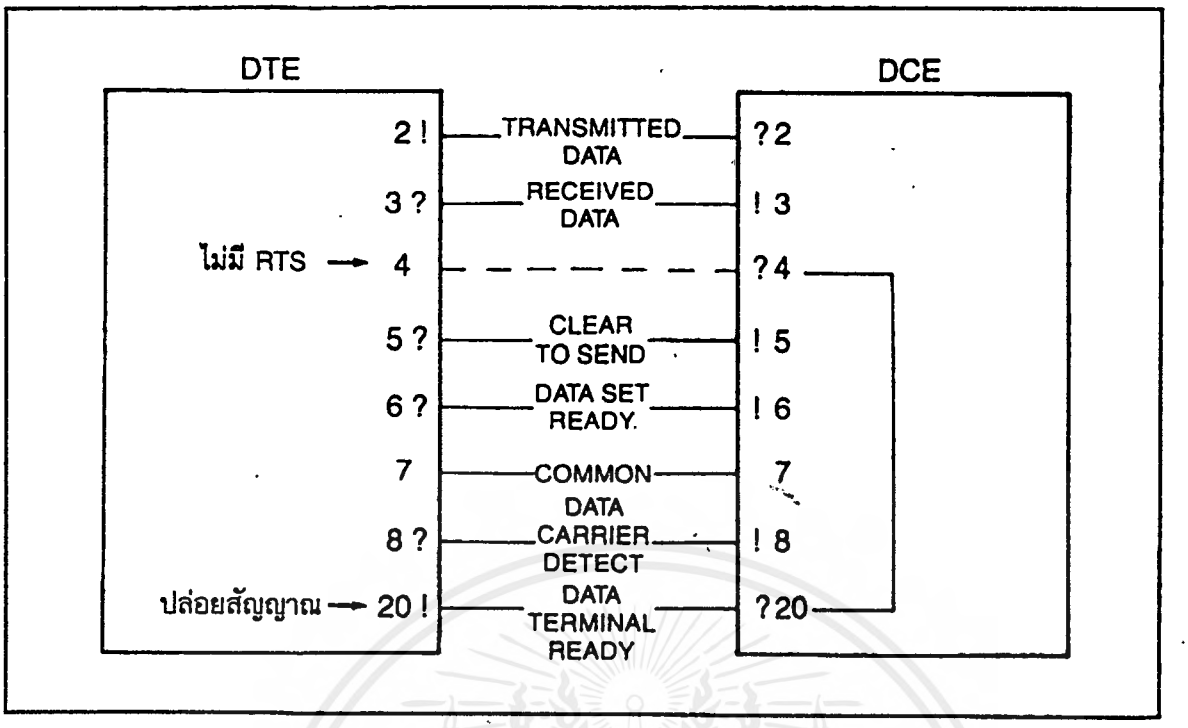
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 8.11 ขาคอนเน็กเตอร์สัญญาณควบคุมขาดหายไป

สมมติเงื่อนไขการทำงานของอุปกรณ์ DCE จะไม่ทำงานหากไม่มีอินพุท RTS เข้ามากระตุ้น แต่อุปกรณ์ DTE กลับไม่มีเอาท์พุท RTS สามารถแก้ไขโดยการยืมสัญญาณอินพุท DTR ของอุปกรณ์ DCE มาใช้ได้ ซึ่งสัญญาณนี้จะมีค่าเป็น " 1 " เมื่ออุปกรณ์ DTE ยืนยันความพร้อมที่จะส่งข้อมูลหรือนั่นก็คือ การแบ่งสัญญาณแรงดันไฟฟ้าในการควบคุมที่จะส่งจากขาคอนเน็กเตอร์ DATA TERMINAL READY ของ DTE ไปป้อนให้อินพุท request to send (rts) ของ DCE ด้วย ดังรูปที่ 8.12

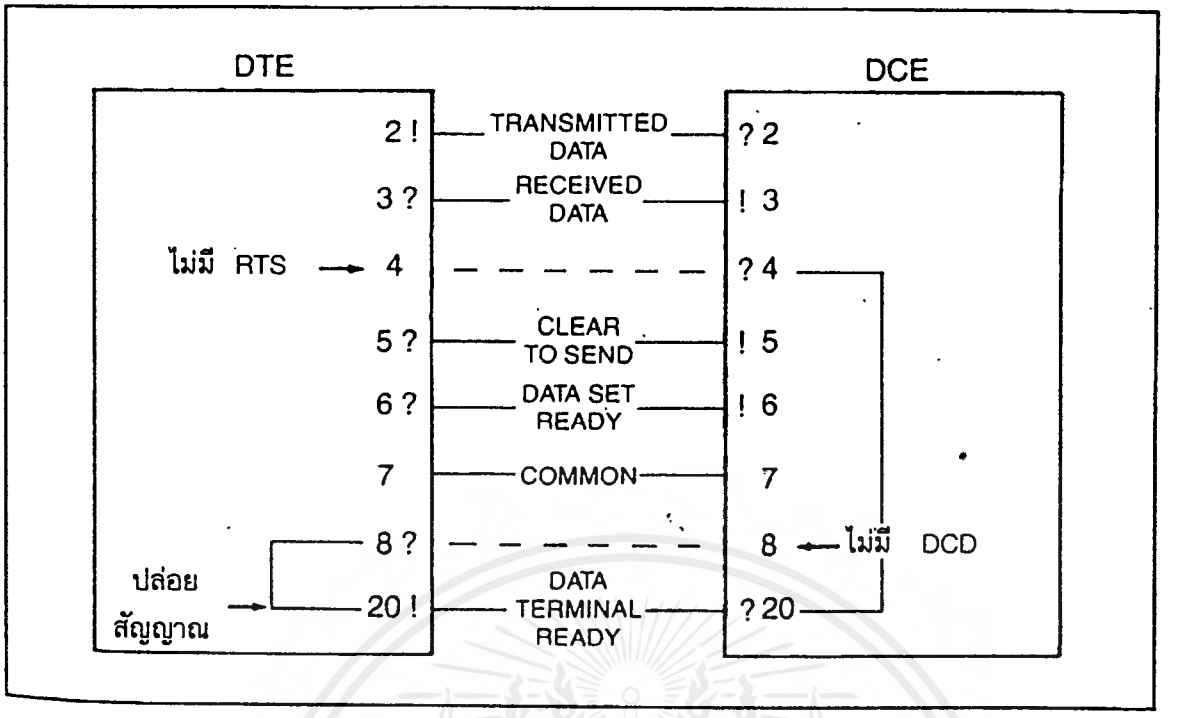
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 8.12 สัญญาณกระตุ้นของขาหนึ่งถูกยืมไปใช้กับอีกขาหนึ่ง

ในทำนองเดียวกันกับปัญหาข้างต้น เครื่องพิมพ์โดยทั่วไป ต้องรอสัญญาณการแฮนด์เชค ที่คอนเน็คเตอร์ขา 8 ก่อนจึงจะทำงาน ถ้าอุปกรณ์ DCE ที่เราจะอินเตอร์เฟสด้วยนั้นเป็นโมเด็ม ปัญหาการจัดหาสัญญาณเอาท์พุท DCD มาทดแทนคงหมดไปเพิ่มเติมจากรูปที่ 8.12 สัญญาณอินพุทที่ต้องการเพิ่ม สามารถนำมาได้จากขาเอาท์พุทอีกขาบนอุปกรณ์ DTE เอง ดังภาพที่ 8.13

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 8.13 การอินเตอร์เฟสอาจต้องดัดแปลงหลายจุด

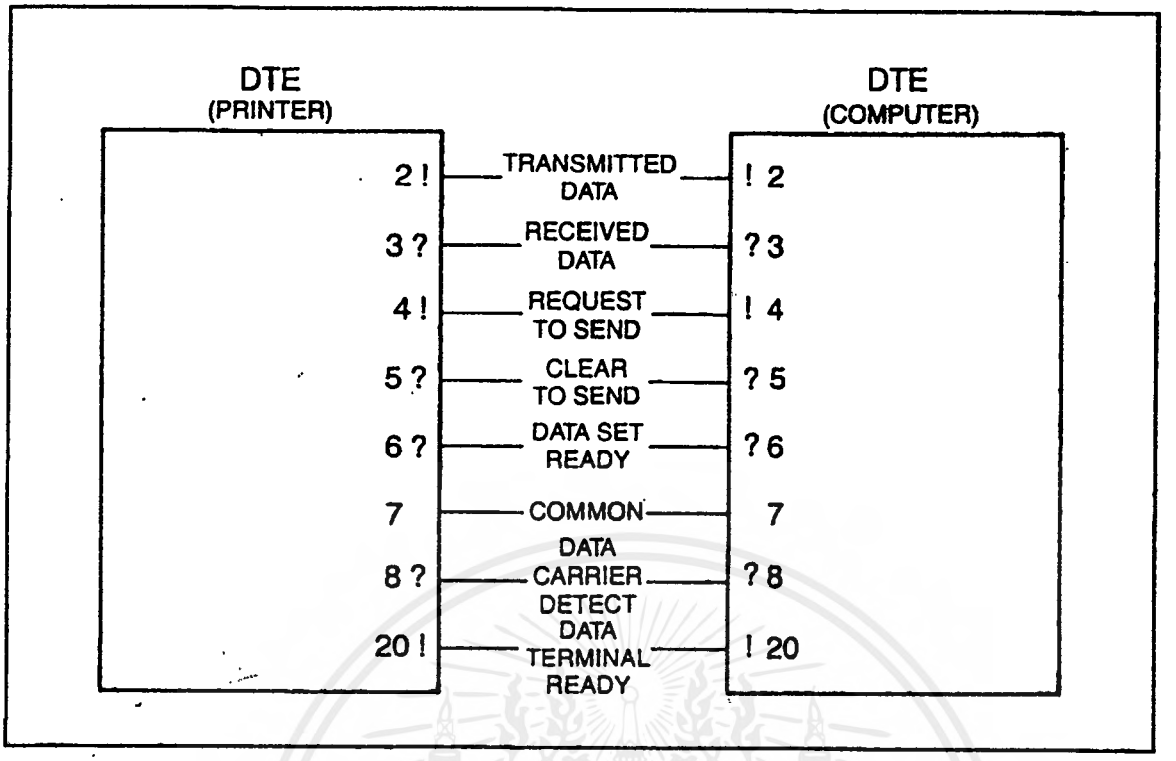
8.2.3 การอินเตอร์เฟสอุปกรณ์ประเภทเดียวกัน (DRE / DTE หรือ DCE / DCE)

การอินเตอร์เฟสอุปกรณ์ที่มีโครงสร้างแบบเดียวกันเข้าด้วยกัน ก็เป็นอีกปัญหาที่พบได้ทั่วไป สมมุติว่าระบบอินเตอร์เฟสของคอมพิวเตอร์เป็นแบบ DTE ก็จะเกิดปัญหาในการนำไปต่อใช้งานกับเครื่องพิมพ์ (DTE)

แต่ถ้าคอมพิวเตอร์ออกแบบมาเป็น DCE ก็จะเจอปัญหาในการนำไปใช้งานกับโมเด็มอีก เว้นแต่ว่าเครื่องคอมพิวเตอร์จะมีพอร์ตการอินเตอร์เฟส 2 พอร์ต สำหรับรองรับการอินเตอร์เฟสทั้งสองแบบ

เพื่อให้มองปัญหาเข้าใจมากขึ้น สมมติว่าคอมพิวเตอร์ที่จะต่อกับเครื่องคอมพิวเตอร์ไม่ได้วางมาเป็นแบบ DCE แต่กลับถูกออกแบบมาเป็น DTE ดังรูปที่ 8.14 จะเห็นว่าเมื่อต่อเครื่องพิมพ์กับคอมพิวเตอร์เข้าด้วยกัน การเชื่อมต่อสัญญาณอินพุทจะต่อกับอินพุท และเอาท์พุทจะต่อกับเอาท์พุท การต่อในลักษณะนี้ ทำให้ไม่มีสัญญาณควบคุมอุปกรณ์ไม่มีการแฮนด์เชค และที่สำคัญ คือ มีการรับส่งข้อมูล จึงจำเป็นต้องเปลี่ยนการวางสายการอินเตอร์เฟสของอุปกรณ์ด้านหนึ่งให้กลับเป็นตรงข้ามกับเดิมที่เป็นอยู่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 8.14 การอินเตอร์เฟสอุปกรณ์ที่มีวงจรอินเตอร์เฟสประเภทเดียวกัน

การปรับระบบการอินเตอร์เฟสแบบกลับเป็นระยะตรงข้ามนี้สามารถทำได้ 3 วิธี คือ

1) เปิดฝาครอบทางด้านหนึ่งหาสายสัญญาณอินเตอร์เฟส แล้วสลับระหว่างคู่อินพุท / เอาท์พุท ซึ่งอุปกรณ์บางชนิดออกแบบให้มีสวิทซ์ภายในสำหรับการสลับระบบการอินเตอร์เฟสโดยเฉพาะ

2) ทำสายเคเบิลอินเตอร์เฟสขึ้นใหม่ โดยสลับคู่อินพุท / เอาท์พุท ที่ปลายด้านหนึ่ง

3) ออกแบบสร้างตัวคอนเน็ตเตอร์ขึ้นมาแก้ปัญหานี้โดยเฉพาะ

การ สลับสาย สัญญาณ คู่อินพุท / เอาท์พุท กรรมวิธีในการแก้ไขให้อินพุทต่อให้ตรงกับเอาท์พุทของมันก็คือ การสลับ (Flipping) ซึ่งอาจจะเป็นการสลับที่สายเคเบิลหรือที่ภายในตัวคอนเน็ตเตอร์ไม่ว่าวิธีใดก็ตาม ผลลัพธ์ที่ได้จะมีความสัมพันธ์เหมือนกัน

8.2.4 หน้าที่การทำงานของแต่ละขา

คำจำกัดความของแต่ละขาบนคอนเน็ตเตอร์นิยามตามหน้าที่ของด้าน DTE เป็นหลัก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขา 1 PROTECTICE GROUND เพื่อป้องกันอันตรายไฟดูดในกรณีที่ตัวจ่ายกำลังไฟฟ้าเกิดการผิดปกติ

ขา 2 TRANSMITED DATA ส่งข้อมูลจาก DTE ไป DCE

ขา 3 RECEIVED DATA ส่งข้อมูลจาก DCE ไป DTE

ขา 4 REQUEST TO SEND เอาท์พุทเอนกประสงค์สามารถนำไปประยุกต์ใช้ได้หลากหลายในโมเด็มแบบ HALF DUPLEX ใช้สัญญาณนี้แสดงความต้องการส่ง

ขา 5 CLEAR TO SEND อินพุทเอนก ประสงค์สามารถนำไปใช้งาน ได้หลากหลายในโมเด็มแบบ HALF DUPLEX สัญญาณนี้ใช้อนุญาติให้ส่งข้อมูลได้

ขา 6 DATA SET READY อินพุทเอนกประสงค์ที่ใช้แล้ว อุปกรณ์ DTE วัดอุปกรณ์ DCE มีไฟเลี้ยวและพร้อมที่จะทำงาน

ขา 7 COMMON จุดอ้างอิงแรงดันสำหรับทุกสัญญาณใน กระบวนการอินเตอร์เฟส (ต้องมี)

ขา 8 DATA CARRIER DETECE สำหรับโมเด็มจะส่งสัญญาณ DCD เมื่อมันรับรู้การติดต่อกับโมเด็มที่อยู่ห่างออกไป สำหรับ DTE สัญญาณ DCD จะถูกนำไปใช้ในการยกเลิกการรับข้อมูล

ขา 20 DATA TERMINAL READY เอาท์พุทเอนกประสงค์ โดยทั่วไปใช้เป็นสัญญาณบอกอุปกรณ์ DCE ว่าอุปกรณ์ DTE ที่อินเตอร์เฟสกันอยู่มีไฟเลี้ยวและพร้อมที่จะทำงาน

ขา 9 , 10 RESERVE FOR TEST

ขา 11 , 18 , 25 UNASSIGN

ขา 12 SECONDARY CARRIER DETECT

ขา 13 SECONDARY CLEAR TO SEND

ขา 14 SECONDARY Tx

ขา 15 ,24 Tx CLOCK

ขา 16 SECONDARY Rx

ขา 17 RECEIVE CLOCK

ขา 19 SECONDARY RTS

ขา 21 SIGNAL QUALITY DETECT

ขา 22 RING INDICATOR

ขา 23 DATA SIGNAL RATE SELECT

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

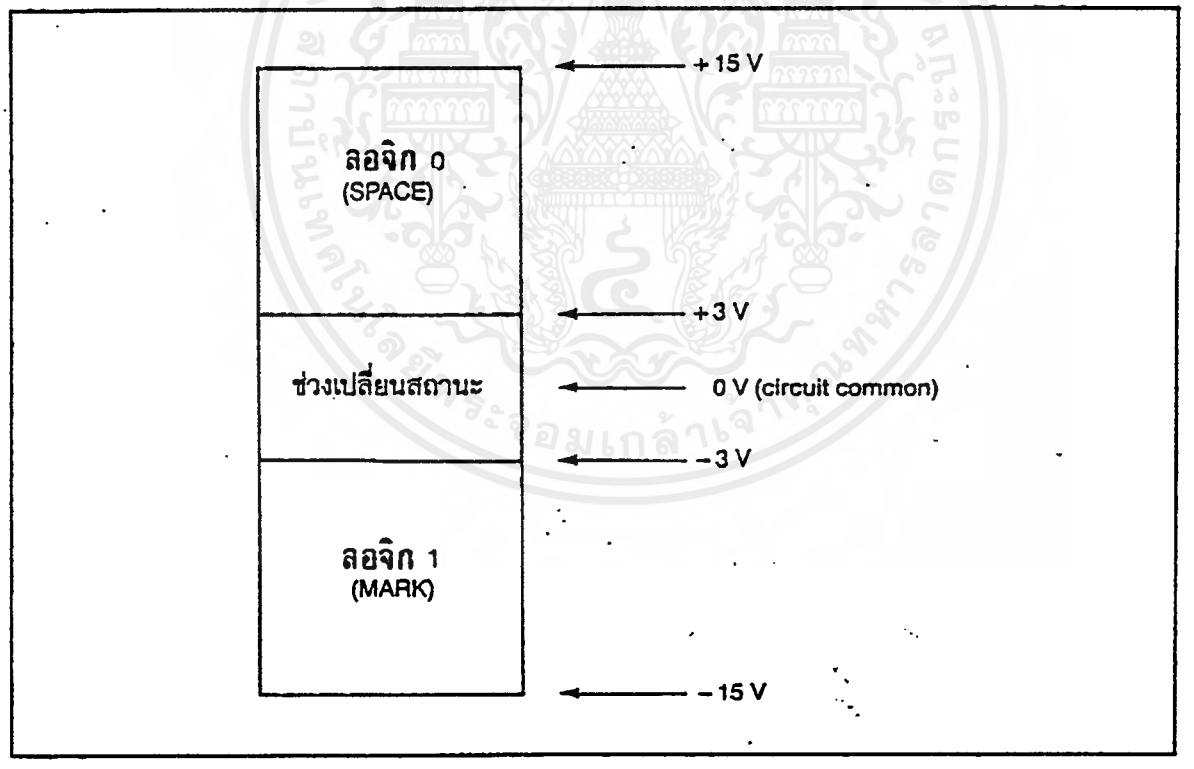
แต่สัญญาณสำคัญต่าง ๆ ที่มีการนำไปใช้เป็นประจำก็มักจะมาจาก 9 ขา เท่านั้น เรามักเรียก กลุ่มของคอนเน็ตเตอร์ขา 1,2,3,4,5,6,7,8 และ 20 ว่า กลุ่ม " BIG ETGHT "

8.3 มาตราฐานสัญญาณไฟฟ้า

มาตราฐาน RS - 232 มีการกำหนดสถานะทางไฟฟ้าเป็นของตัวเอง โดยแรงดันไฟที่ใช้งานจะอยู่ในช่วง + 25 ถึง -25 V โดยขึ้นอยู่กับสภาวะต่าง ๆ กัน โดยปกติอุปกรณ์คอมพิวเตอร์มักจะนิยามคำของ ลอจิกจากขนาดของแรงดัน แต่สำหรับในมาตราฐาน RS - 232 กำหนดเป็นระดับสัญญาณลอจิกแทน หมายความว่าแรงดันที่เป็นบวก และลบจะเป็นตัวกำหนดสัญญาณลอจิกนั้น ๆ

8.3.1 คำจำกัดความของสัญญาณลอจิก

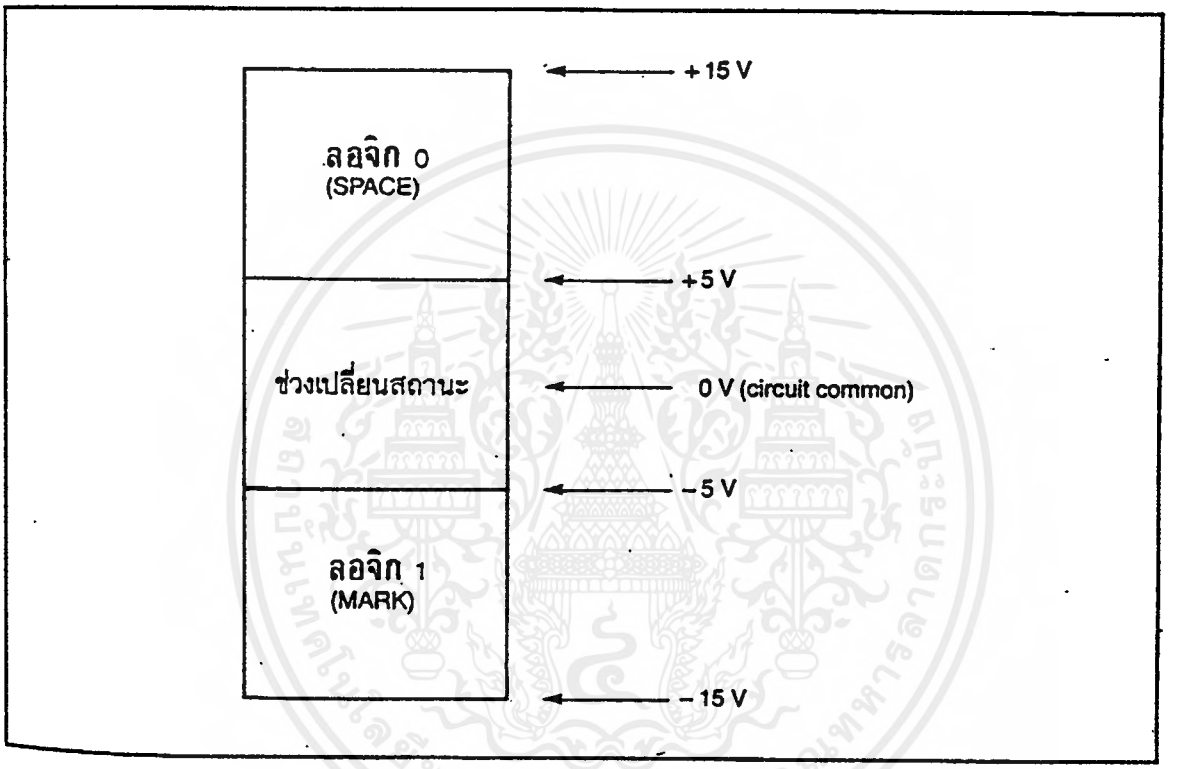
การส่งข้อมูลจากวงจรอินเตอร์เฟส แรงดัน บวกจะแทนด้วยลอจิก " 0 " ในขณะที่แรงดันลบ แทนด้วย " 1 " หากต้องการที่จะใช้การอินเตอร์เฟส เพิ่มเติมต้องเข้าใจถึงรายละเอียดของความสัมพันธ์นี้ในรูปที่ 8.15



รูปที่ 8.15 คำจำกัดความสัญญาณลอจิกที่เอาท์พุท RS - 232

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถ้าสังเกตสัญญาณลอจิกที่กลับกันให้ดี คือ แรงดันลบแทนด้วย ลอจิก " 1 " และแรงดันบวกแทนด้วยลอจิก " 0 " เพื่อให้แน่ใจว่า " 0 " แรงดันไฟที่เอาท์พุทจะต้องอยู่ในช่วง + 5 V ถึง +15 V ในทำนองเดียวกันลอจิก " 1 " แรงดันไฟที่เอาท์พุทอยู่ในช่วง - 5 V ถึง - 15 V สำหรับช่องว่างหรือ dead-band ที่อยู่ในช่วง + 5 V ถึง - 5 V เรียกว่า ช่วงเปลี่ยนสถานะ (Transition Region) เป็นจุดที่ไม่สามารถกำหนดสัญญาณลอจิกได้ซึ่งหมายความว่า แรงดันเอาท์พุทในช่วง + 5 V ถึง - 5 V อาจเป็นได้ทั้งลอจิก " 0 " หรือ " 1 " ก็ได้ รูปที่ 8.16. แสดงคำจำกัดความของสัญญาณลอจิกที่อินพุท



รูปที่ 8.16 คำจำกัดความสัญญาณลอจิกที่อินพุทของ RS - 232

ข้อแตกต่างระหว่างคำจำกัดความสำหรับอินพุทกับเอาท์พุทคือ ความกว้างของช่วงเปลี่ยนสถานะ (Transition Region) โดยช่วงที่ไม่สามารถกำหนดสัญญาณ ลอจิกได้ของอินพุทว้าง 6V (จาก +3 V ถึง -3 V) ในขณะที่เอาท์พุทกว้างถึง 10 V (จาก + 5 V ถึง - 5 V) ซึ่งความแตกต่างนี้มีความสัมพันธ์มากทีเดียว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8.3.2ช่วงการยอมรับสัญญาณรบกวน

ความแตกต่างระหว่างคำจำกัดความของแรงดันต่ำสุด ที่วงจรต่ำสุดยอมรับได้ เรียกว่า ช่วงการยอมรับสัญญาณรบกวน (Noise Maragin) หมายความว่า วงจรยอมให้มีสัญญาณรบกวนออกจากเอาท์พุทสู่อินพุทได้โดยไม่ส่งผลต่อสัญญาณลอจิกที่อินพุท ซึ่งคุณสมบัติข้อนี้มีประโยชน์มากในเวลาที่จำเป็น ต้องเดินสายข้อมูลผ่านอุปกรณ์ที่เป็นตัวสร้างสัญญาณรบกวนเช่น มอเตอร์ไฟฟ้า หลอดไฟฟ้า ฟลูออเรสเซนต์ วงจรหรี่ไฟ และอุปกรณ์การสื่อสารต่าง ๆ

ส่วนต่างระหว่างช่วงเปลี่ยนสถานะอินพุทและ เอาท์พุทนอกจากจะทำหน้าที่เป็นช่วงยอมรับสัญญาณรบกวนแล้วยังทำหน้าที่เป็นช่วงปลอดภัย (Safety Margin) ด้วยโดยการให้แรงดันเผื่อสำหรับแรงดันที่ตกคร่อมสายเคเบิล ทำให้วงจรสามารถรับแรงดันที่ลดลงจากเอาท์พุทได้ถึง 2v โดยข้อมูลไม่ตกเข้าสู่ช่วงที่กำหนดสัญญาณลอจิกไม่ได้ของอินพุท

เนื่องจากแรงดันไฟกระแสตรง (Direct Current Voltage) สูญเสียไปน้อยมาก ในสายเคเบิลจนสามารถตัดทิ้งไปได้ แม้ในสายขนาด ยาว ๆ ดังนั้น มาตราฐาน RS - 232 จึงมีข้อกำหนดสำหรับสัญญาณควบคุม น้อยกว่าสัญญาณข้อมูลเนื่องจากสัญญาณควบคุม และสัญญาณแฮนด์เชคเป็นสัญญาณแรงดันไฟฟ้ากระแสตรง

8.3.2ความเร็วและระยะทางในการส่งข้อมูล

ในขณะที่ความเร็วในการส่งข้อมูลเพิ่มขึ้น แรงดันไฟที่ตกคร่อม ตัวเก็บประจุและตัวเหนี่ยวนำบนสายเคเบิลจะมีผลต่อข้อมูลมากขึ้น แรงดันไฟที่ลดลงนี้เป็นที่รู้จักกันดีว่าเป็นผลของความถี่สูง (High Frequency Effect) จะแปรผันโดยตรงกับความยาวของสายเคเบิล ด้วยเหตุ ที่ระดับของสัญญาณลดต่ำสูงเนื่องจากการสูญเสีย แรงดันไฟนี้ ทำให้ความยาวของสายเป็นตัวกำหนดความเร็วในการส่งข้อมูลโดยตรง

ใช้สายเคเบิลได้ยาวเท่าไรนั้น EIA ได้จำกัดค่าความจุรวมของสายเคเบิลไว้ที่ 2500 พิโกฟาร์ครัด เนื่องจากค่าเฉลี่ยความจุของสายเคเบิลคือ 40 - 50 พิโกฟาร์ครัด ต่อฟุต ดังนั้นสายเคเบิลที่ยาวที่สุดที่เป็นไปได้คือ ประมาณ 50 ฟุต

ในการทดลองใช้สายชนิด 3 ตัวนำ 22 AWG ยาว 250 ฟุต จำนวน 11 เส้น และไม่มีการป้องกันสัญญาณรบกวนนำมาต่อเข้ากับพอร์ต RS - 232 ของคอมพิวเตอร์จากนั้นก็เขียนโปรแกรมให้ส่งรหัส แอสกี ตัวอักษร U โปรแกรมจะทำการนับข้อผิดพลาดที่เกิดขึ้นในขณะที่ข้อมูลถูกส่งออกมาอย่างต่อเนื่อง ผลการทดลองได้บันทึกความยาว สายสูงสุดที่ระบบสามารถส่งตัวอักษรมาประมาณ 1,000,000 ตัวอักษรโดยไม่มีข้อผิดพลาดเลย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อัตราเร็ว	ความยาวสาย (ฟุต)
110	2750
300	2500
600	2500
1200	1750
2400	750
4800	500
9600	250
19200	< 250

ความสัมพันธ์ในตารางนี้ได้ทดลองไว้เมื่อปี 1983 และ ในการทดสอบครั้งล่าสุด สามารถส่งข้อมูลได้ 20 ล้านตัว อักษรที่อัตราเร็ว 9,600 โดยใช้สาย 1,000 ฟุต และที่อัตราเร็ว 19,200 สำหรับสาย 750 ฟุต จะเห็นได้ว่า ประสิทธิภาพการส่งดีขึ้นเป็นผลมาจากการพัฒนาของฮาร์ดแวร์อย่างไม่ต้องสงสัยโดยเฉพาะ อย่างยิ่งการพัฒนาในวงจรรวมที่ใช้เป็น Line Driver และ Line Receiver ใน RS - 232

8.3.4 คุณสมบัติโดยย่อของมาตราฐาน RS - 232

- Driver output logic Levals with 3k to 7k load	15 V > Logic 0 > 5 V
	- 5 V > Logic 1 > 15 V
- Driver output voltage when open circuit	Vo > 25 V
- Driver output impedance with power off	Ro > 300 ohms
- Output short circuit current	Io < 0.5 A
- Driver slew rate	dv /dt < 30 V/ s
- Receiver input impendance	7 k > Rin > 3 k
- Receiver input voltage	15 V compatibel with Driver
- Receiver output with open circuit input	MARK
- Receiver output with + 3 V input	SPACE

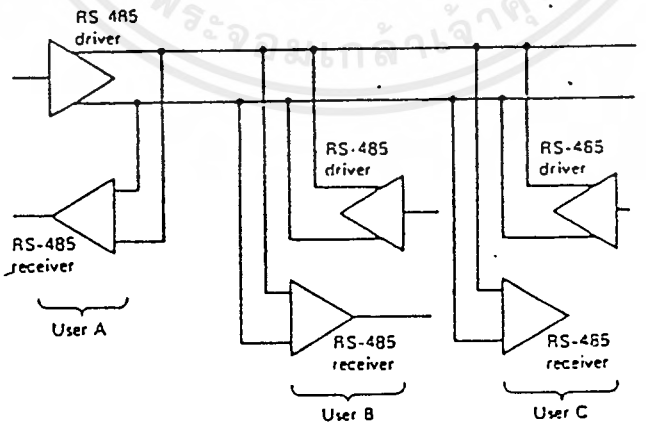
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Recevier output with - 3 V input MARK
- +5 V to +15 V Logic 0 = SPACE = CONTROL ON
- +3 V to +5 V or -5 V to -3 V Noise Margin
- -15 V to -5 V Logic 1 = MARK = CONTROL OFF

8.4.มาตรฐาน RS - 485

มาตรฐาน RS - 485 สามารถดัดแปลงเพื่อให้เหมาะสมกับงานได้และมีประสิทธิภาพการทำงานสูงสุดในบรรดามาตรฐาน RS ทั้ง 5 ชนิด ซึ่งจะใช้การรับส่งสัญญาณ ในลักษณะ Differential ซึ่งเหมือนกับ RS - 422 A ปัจจัยที่สำคัญของ RS - 485 ก็คือ เป็นระบบการสื่อสารข้อมูลแบบ Multidrop ที่สมบูรณ์แบบที่สุด จำนวนตัวส่ง สูงสุด 32 ตัว และจำนวนตัวรับสูงสุด 32 ตัว จะถูกเชื่อมต่อกัน โดยใช้สายเพียงคู่เดียวเท่านั้น รายละเอียดของสัญญาณที่ใช้ในการรับส่ง การทำงานในโหมด Highimpedance ของตัวส่งใน RS - 485 จะคล้ายกันกับระบบไฟฟ้าใน RS - 422 A

รูปที่ 8.17 แสดงรูปแบบของ RS - 485 แบบเบื้องต้น ระดับแรงดันที่ใช้จะแตกต่าง RS - 422 A เพียงเล็กน้อย สัญญาณ + 1.5 V จะแทนด้วย เลขไบนารี่ 0 และระดับสัญญาณ -1.5 V จะแทนด้วยเลขไบนารี่ 1 ตัวรับจะใช้ระดับสัญญาณ เพียง + 200 mv และ - 200 mv เพื่อนำมาพิจารณาว่าเป็นระดับสัญญาณอะไร การใช้สายเส้นเล็กจะเป็นการช่วยลด Noise ถึงแม้ว่าสัญญาณจะมีขนาดเล็ก แต่ก็เพียงพอที่จะนำมาพิจารณาระดับของสัญญาณสำหรับการใช้งานโดยทั่วไป มีอัตราการรับส่งข้อมูลสูงสุด 10 เม็กกะบิต ต่อวินาที



รูปที่ 8.17 รูปแบบมาตรฐาน RS - 485

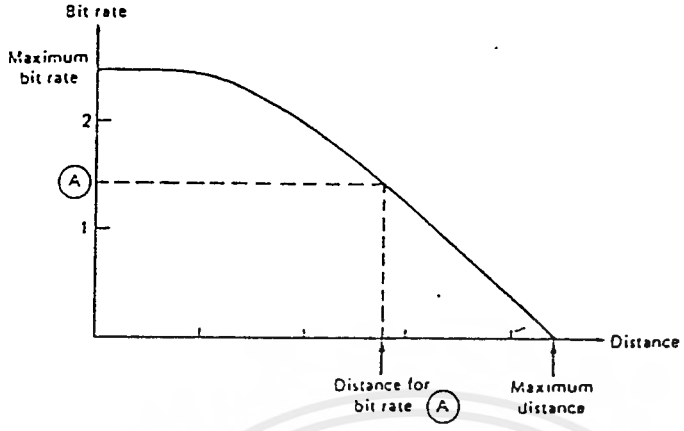
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มาตราฐาน RS แต่ละแบบจะนำไปประยุกต์เพื่อให้เหมาะสมกับงาน ตารางที่ เป็นการเปรียบเทียบระหว่าง RS - 232 , RS - 423 A , RS - 422 A และ RS- 485 ไม่มีมาตราฐานตัวใดที่สามารถรับส่งสัญญาณได้ในระยะทางสูงสุด และอัตราบิตสูงสุดได้ในเวลาเดียวกัน จากกราฟในภาพที่ แสดงอัตรารับส่งสัญญาณสูงสุดกับระยะทาง จะเห็นได้ว่า เมื่อระยะทางเพิ่มขึ้นก็จะเกิดการ Drop ของอัตรารับส่งเพิ่มขึ้นตามระยะทาง

Feature	RS-232	RS-423	RS-422	RS-485
Mode	Single-ended	Single-ended	Differential	Differential
No. of drivers and receivers	1 Driver 1 Receiver	1 Driver 10 Receivers	1 Driver 10 Receivers	32 Drivers 32 Receivers
Max. distance (ft)	50	4000	4000	4000
Max. data rate (bits/s)	20K	100K	10M	10M
Signal levels (V)	3 to 25 −3 to −25	3.6 to 6 −3.6 to −6	2 to 6 −2 to −6	1.5 to 6 −1.5 to −6
Receiver decision point (V)	+3 −3	0.2 −0.2	0.2 −0.2	0.2 −0.2

ตารางที่ 1 แสดงการเปรียบเทียบระหว่าง RS - 232, RS - 423 A , RS - 422 A และ RS - 485

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

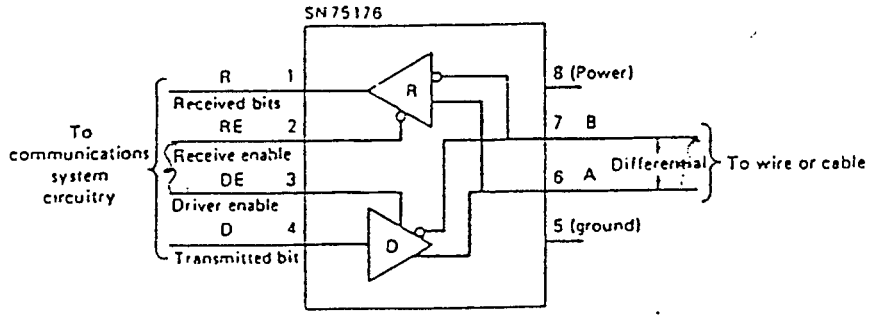


รูปที่ 8.18 กราฟแสดงอัตราการส่งข้อมูลกับระยะทางที่เปลี่ยนไป

8.4.2 การทำงานของRS - 485 (IC เบอร์ SN 75176)

IC เบอร์ SN 75176 เป็นของ Taxas Instrument ใช้ differential แบบ 3 - S tate ตัวหนึ่งเป็นตัวส่งและอีกตัวหนึ่งเป็นตัวรับเป็น IC 8 ขา ฟังก์ชั่น การทำงานจะแสดงดังรูปที่ 8.19

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Input D	Enable DE	Outputs A	Outputs B
1	1	H	L
0	1	L	H
X	0	Z	Z

Differential Inputs A−B	Enable RE	Output R
A−B > 0.2 V	0	1
A−B < −0.2 V	0	unknown
A−B between −0.2 and 0.2	0	0
X	1	Z

รูปที่ 8.19 ลักษณะโครงสร้างภายในและฟังก์ชั่นการทำงานของ SN 75176

เพื่อให้เข้าใจการทำงานของ IC เบอร์ SN 75176 ตัวส่งบิตข้อมูลของตัว IC จะต่อจากขา 4 คือ ขา D (Driver) สายสัญญาณ Differential จะต่อที่ขา 6 และ 7 เรียกว่า ขา A และ B ข้อมูลจะถูกเปลี่ยนเป็นสัญญาณแรงดันเป็นฟังก์ชั่นภายในตัว IC เมื่อข้อมูลถูกส่งออกมาเป็น " 1 " ที่ เอาท์พุท A จะมีแรงดันสูงและที่เอาท์พุท B จะมีแรงดันต่ำเมื่อข้อมูลเป็น " 0 " แรงดันระหว่าง A กับ B ก็จะเป็นตรงข้ามกับ คือ แรงดันที่ B จะมากกว่า A

การใช้งานของตัวส่งสามารถใช้ในโหมด High - Impedance 3 - State ซึ่งจะใช้ขา Driver Enabable (DE) การทำงานในโหมดนี้ เมื่อขา DE เป็น 0 เอาท์พุท A และ B ก็จะเข้าสู่โหมด 3 - State โดยจะไม่คำนึงถึงสถานะของข้อมูลที่ขา D ฟังก์ชั่น การทำงานก็จะแสดงในภาพที่

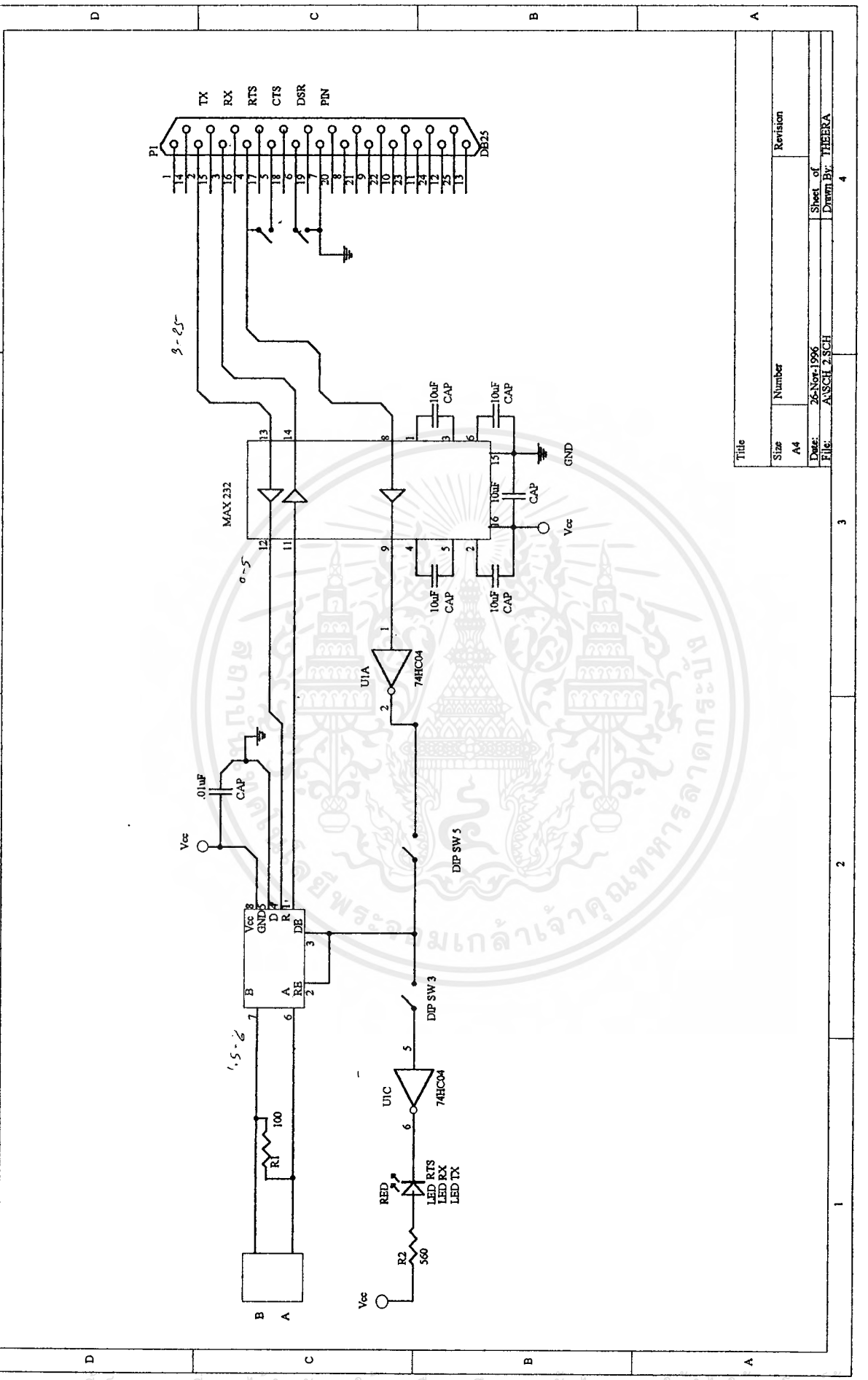
ฟังก์ชั่นการทำงานของตัวรับจะแสดงดังภาพที่ เมื่อมีข้อมูลเข้ามาทางขา A และ B เมื่อแรงดันที่ขาทั้งสองต่างกันมากกว่า 0.2 V ที่เอาท์พุท R จะมีค่าเป็น " 1 " ในทางกลับกันเมื่อแรงดันที่ A และ B ต่างกันมากกว่า -0.2 V (B มากกว่า A) ที่เอาท์พุท R

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

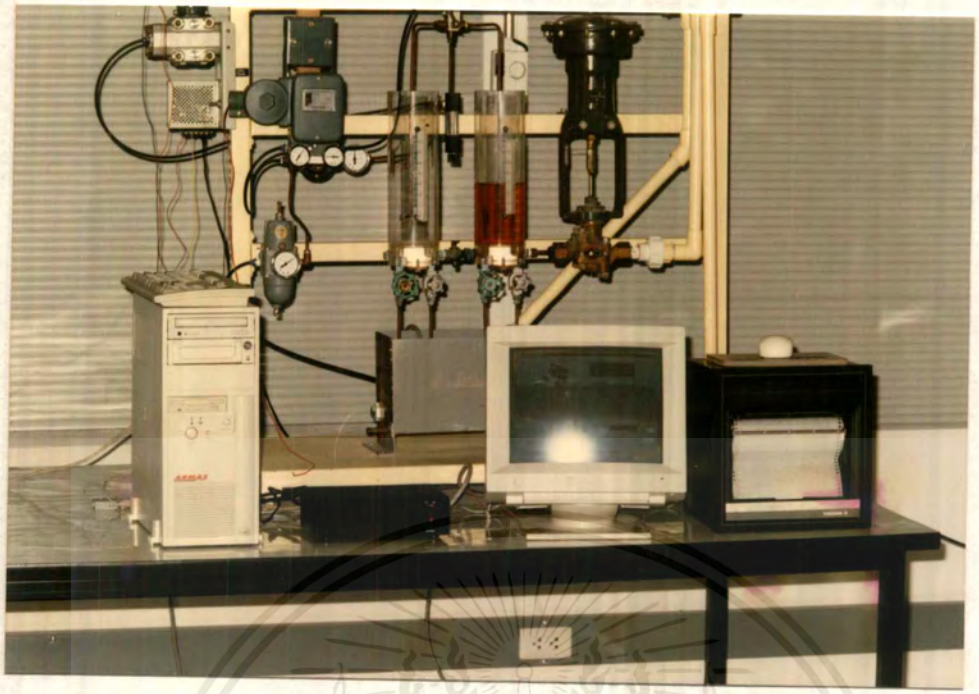
ก็จะมีค่าเป็น " 0 " แต่ถ้าค่าความแตกต่างของ A กับ B น้อยกว่า 0. 2 V จะทำให้หาค่าไม่ได้ที่เอาท์พุท R อาจเป็นได้ทั้ง " 0 " หรือ " 1 "

ตัวรับก็เหมือนกับตัวส่ง ซึ่งสามารถใช้งานในโหมด 3 - State ซึ่งอาจจะใช้กับบางวงจรขา 2 (Receiver Enable (RE)) ของ IC จะเป็นตัวควบคุมโหมดเมื่อขา RE เป็น " 1 " เอาท์พุท R ก็จะเป็น High - Impedance ซึ่งไม่ต้องคำนึงถึงอินพุท A และ B แต่ถ้า RE เป็น " 0 " ที่เอาท์พุทก็อาจจะ " 0 " หรือ " 1 " ซึ่งขึ้นอยู่กับอินพุท A และ B

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปแสดง (232 TO 485 CONVERTER) กับชุดการควบคุมกระบวนการระดับน้ำ

รูปแสดงการต่อวงจรภายในของ 232 TO 485 CONVERTER

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 9

Digital I/P , O/P Unit

9.1 การออกแบบ 8031 ไมโครคอนโทรลเลอร์บอร์ด

ปกติแล้ว IC 8031 (ไม่มีรอมภายใน) ไม่สามารถทำงานตามลำพังได้ ซึ่งตัวเองต้องการวงจรของหน่วยความจำภายนอก แต่ถ้าโปรแกรมที่ใช้งานต้องการพื้นที่หน่วยความจำไม่มากก็สามารถใช้พื้นที่หน่วยความจำภายในตัว IC 8031 ได้ และต้องต่อวงจร I/O และส่วนอินเตอร์เฟสต่าง ๆ ที่นำมาใช้งานโดยเฉพาะในการออกแบบบอร์ด 8031 จำเป็นต้องพร้อมที่จะร่วมวงจรเหล่านั้นไปด้วย ซึ่งการออกแบบนั้นต้องทำตามแนวทางวิธี ซึ่งสามารถประหยัด เกิดประโยชน์และยืดหยุ่นในการใช้ได้

การถอดรหัสแอดเดรสหน่วยความจำ

หน่วยความจำภายนอกถูกใช้ร่วมกับพอร์ตเบอร์ศูนย์ เป็นดาต้าบัสและแอดเดรสไบร์ต่ำ และพอร์ตเบอร์สองเป็นแอดเดรสบัสไบต์สูง ดาต้าและแอดเดรสไบท์ต่ำถูกทำมัลติเพล็กซ์บนพอร์ตเบอร์ศูนย์ ตัวแลทช์แอดเดรสภายนอกเบอร์ 74HC373E ถูกต่อเข้ากับพอร์ตเบอร์ศูนย์ เก็บค่าแอดเดรสไบต์ต่ำ เมื่อไรก็ตามจะเข้าถึงตำแหน่งของหน่วยความจำภายนอก โดยที่ 8031 จะส่งพัลซ์ ALE ไปทริกเพื่อให้ 74HC373E แลทช์แอดเดรส และพอร์ตเบอร์ศูนย์ ยังเป็นบัสแบบสองทิศทางในระหว่างช่วงคาบไซเคิลอ่านและเขียน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

9.2 พอร์ตแบบขนาน

พอร์ตแบบขนานที่สามารถใช้งานได้มีจำนวน 4 พอร์ต คือ พอร์ต คือ 0,1,2,3 ซึ่งเป็นพอร์ตขนาด 8 บิท ทั้งลักษณะทำงานได้ทั้งลักษณะของสัญญาณเดี่ยว ๆ หรือกลุ่มของสัญญาณได้ นอกจากนี้พอร์ต 0, 2 และ 3 ซึ่ง สามารถนำไปใช้งานอื่น ๆ ที่ไม่ใช่เป็นพอร์ตอินพุต/เอาท์พุตได้ โดยพอร์ต 0 จะทำหน้าที่มัลติเพล็กซ์ ระหว่างบัสแอดเดรสไบต์ต่ำและบัสข้อมูลสำหรับการติดต่อกับวงจรประกอบร่วมกับข้อมูลบัสแอดเดรสไบต์สูงซึ่งจะส่งออกมาทางพอร์ต 2 สำหรับพอร์ต 3 สามารถนำไปเป็นขาสัญญาณของการอินเตอร์รัปต์ต่าง ๆ ซึ่งรวมทั้งการสร้างสัญญาณควบคุม RD\ และ WR\ เพื่อทำหน้าที่อ่านหรือเขียนหน่วยความจำข้อมูลภายนอกด้วย

9.3 การใช้งานพอร์ตเป็นการอินพุท

การใช้งานพอร์ตเป็นการอินพุทข้อมูลจะต้องเริ่มด้วยการส่งข้อมูลที่มีค่าเป็น 1 ออกมาทางบิทของพอร์ตนั้นก่อนเป็นลำดับแรก เพื่อหยุดการทำงานของทรานซิสเตอร์ที่ทำหน้าที่รับสัญญาณเอาท์พุทของบิทนั้น ทำให้ขาสัญญาณของบิทถูกต่อเข้ากับตัวต้านทานซึ่งทำหน้าที่ Pull-Up ภายในซึ่งมีผลให้บิตนั้น ๆ ของพอร์ต 1,2,3 เป็นสภาวะลอจิกสูง ตัวต้านทานนี้มีค่าประมาณ 50K โอห์ม ซึ่งมีค่าสูงมากและทำให้อุปกรณ์ภายนอกสามารถ ขับสัญญาณของพอร์ตเหล่านี้เป็นลอจิกต่ำได้ง่าย สำหรับบิทของพอร์ต 0 นั้น แม้ว่าจะมีหลักการทำงานที่คล้ายคลึงกันกับบิทของพอร์ตอื่น ๆ แต่เนื่องจากการที่ไม่มีตัวต้านทานทำหน้าที่ Pull-Up ภายในไว้ทำให้เมื่อทรานซิสเตอร์ที่ทำหน้าที่ขับสัญญาณเอาท์พุทนั้นหยุดการทำงานก็จะเป็นผลให้ขาสัญญาณนี้อยู่ในสภาวะอิมพิแดนซ์สูงแทน

9.4 การใช้งานพอร์ตเป็นการเอาท์พุท

เมื่อมีการส่งข้อมูลที่มีค่าเป็น 0 ให้กับแต่ละบิทของพอร์ตข้อมูลนี้จะถูกส่งให้กับฟลิปฟลอปซึ่งจะค้างค่านี้ไว้ และมีผลทำให้ทรานซิสเตอร์ที่ทำหน้าที่ขับสัญญาณเอาท์พุทนั้นทำงาน ดังนั้นสัญญาณก็จะมีสภาวะลอจิจเป็นลอจิกต่ำด้วย

ส่วนการส่งข้อมูลที่มีค่าเป็น 1 ออกมานั้น ในกรณีที่เป็นการทำงานในแต่ละบิทของพอร์ต 1,2 หรือ 3 จะทำให้ทรานซิสเตอร์ที่ทำหน้าที่รับสัญญาณเอาท์พุทนั้นหยุดการทำงาน มีผลทำให้ขาสัญญาณเป็นลอจิกสูงด้วย ตัวต้านทานที่ Pull-Up อยู่ภายในนั้น แต่สำหรับการทำงานในแต่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ละบิททางพอร์ต 0 นั้น จะมีผลที่แตกต่างออกไป โดยขาสัญญาณจะเป็นสภาวะอิมพิแดนซ์สูงแทน เนื่องจากไม่มีตัวต้านทานภายในเชื่อมต่ออยู่นั่นเอง ดังนั้นในการงานพอร์ต 0เป็น การเอา? พุทข้อมูลจึงจำเป็นต้องใช้ตัวต้านทานภายนอก Pull-Up สัญญาณไว้กับลอจิกสูงแทน

9.5 รายละเอียดต่าง ๆ บน Main Board

ไมโครคอนโทรลเลอร์

บน Main Board จะใช้ไมโครคอนโทรลเลอร์เบอร์ 80C31 เป็นหลัก แต่อย่างไรก็ตามผู้ใช้สามารถใช้กับไมโครคอนโทรลเลอร์เบอร์ต่าง ๆ ในตระกูล MCS-51 ที่เป็นแบบ 40-PIN DIP ได้ทั้งหมด เช่น 8032 8751 8752 ซึ่งจะทำให้ได้คุณสมบัติเป็นไปตามโครงสร้างของเบอร์นั้น ๆ การเลือก JUMPER /EA จะใช้เพื่อการเลือกให้ทำงานจาก ROM หรือ EPROM ภายในตัวไมโคร (INT) หรือเลือกจาก EPROM ภายนอก (EXT) ทั้งนี้การเลือก /EA ในตำแหน่ง INT จะใช้กับไมโครที่มีโปรแกรมอยู่ภายในเท่านั้น ซึ่งปกตอจะเป็น 8751 หรือ 8752 (กรณี 8051 หรือ 8052 มี ROM ภายในอยู่ก็จริง แต่ในทางปฏิบัติ การโปรแกรม ROM ภายในของ 8051 หรือ 8052 จะทำได้จากโรงงานผู้ผลิตเท่านั้น และรับจำนวนมาก ๆ เพราะฉะนั้นในทางปฏิบัติไม่ควรจะมีเบอร์ 8051 หรือ 8052 ขายอยู่ในท้องตลาดเลย แต่สำหรับบ้านเราอาจจะเป็นไปได้ ทั้งนี้อาจจะเป็นการแกบมาจากบอร์ดอื่น ๆ

หน่วยความจำและการเลือก JUMPER

หน่วยความจำบนบอร์ดสามารถใช้ได้ทั้งแบบ ROM (EPROM) และ RAM ซึ่งในด้านไมโครก็จะมองได้เป็น PROGRAM และ DATA ทั้งนี้แล้วแต่เบอร์ไมโครที่เลือกใช้ และยังสามารถเลือกเบอร์ต่าง ๆ ได้ตามต้องการ โดยการเลือก JUMPER ทั้ง 2 ชุด ที่ด้านขวามือของ SOCKET หน่วยความจำ ซึ่งจะสรุปได้ดังต่อไปนี้

เบอร์ที่ใช้	ลักษณะการมองหน่วยความจำ		JUMPER ชุดแรก
JUMPERชุดที่สอง			
27C64	PROGRAM	ROM	8,16K
27C128	PROGRAM	ROM	8,16K
27C256	PROGRAM	ROM	32K
6264	DATA	RAM	8,16K
62256	DATA	RAM	32K

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การเลือก JUMPER นี้ จะกระทำได้โดยปรับตัว JUMPER ให้ตรงกับช่องที่ระบายที่ที่ขาวทึบตรงตาม COLUMN ที่ต้องการ เช่น ถ้าต้องการเลือก JUMPER ชุดแรกให้อยู่ที่ตำแหน่ง ROM ก็ให้ใส่ตัวตัว JUMPER แบบตัวเว้นตัวโดยชิดไปทางด้านนั่นเอง

SERIAL PORT

พอร์ทการสื่อสารอนุกรมของบอร์ด สามารถเลือกใช้ได้ 2 แบบคือ RS232 และ RS485 โดยถ้าต้องการใช้เป็น RS 232 ก็ให้เสียบชิพเบอร์ MAX232 และใช้งานที่ขั้วต่อแบบ 3 PIN แต่ถ้าต้องการใช้เป็น RS485 ก็ให้เสียบชิพเบอร์ 75176 และใช้งานที่ขั้วต่อแบบ 2 PIN โดยในกรณีของ RS485 นี้ จะใช้ขา T0 ของตัวไมโคร เป็นตัวควบคุมทิศทางการรับและส่ง โดยถ้า T0 = 0 จะเป็นการรับข้อมูล และ T0 = 1 จะเป็นการส่งข้อมูล RS485 จะมีแนวทางการใช้งานในทำนองเดียวกับ RS232 แต่จะแตกต่างในด้านของระดับแรงไฟ คือจะใช้ที่ 5V โดยจะเป็นบวกและลบสลับกันตามค่าบิท 0 และ 1 ซึ่งจะยังผลให้มีการหักล้างกันเอง ทำให้ลดสัญญาณรบกวนได้เป็นอย่างดี และ RS485 ยังมีการสื่อสารในแบบ HALF DUPLEX โดยจะต้องรับส่งข้อมูลคนละจังหวะไม่สามารถสวนทางกันได้แบบ RS232 จึงทำให้ต้องมีบิทเพื่อการควบคุมการรับส่งนั่นเอง และด้วยคุณสมบัตินี้เอง ที่ทำให้ RS485 สามารถต่อเป็นระบบ NETORK ได้ซึ่งจะต่อจำนวน NODE ได้ถึง 32 จุด และมีระยะทางได้ถึง 1.2 Km อีกด้วย จุดสำคัญของระบบ NETWORK ก็คือการทำโปรแกรมควบคุม (PROTOCAL) นั้นเอง โดยจะต้องควบคุมทิศทางและการสื่อสารข้อมูลของแต่ละจุดในระบบ ให้ทำงานได้อย่างเป็นระเบียบและมีประสิทธิภาพที่สุด

SYSTM BUS และ PORT 1 BUS

SYSTEM BUS ของบอร์ด จะใช้ขาสัญญานจากตัวไมโคร เป็นขนาด 40 PIN ซึ่งใช้สำหรับการขยายระบบตามต้องการ หรือจะใช้กับบอร์ดขยายก็ได้ ส่วน PORT1 BUS ก็สามารถนำไปใช้เพื่อการประยุกต์ใด ๆ ได้ ทั้งนี้ PORT1 BUS ก็คือบางส่วนจาก SYSTEM BUS โดยทำไว้เพื่อให้ใช้งานได้อย่างสะดวกยิ่งขึ้น สามารถทำหน้าที่เป็น I/P หรือ O/P ก็ได้ ตามแต่ วัตถุประสงค์ที่เราจะนำไปใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

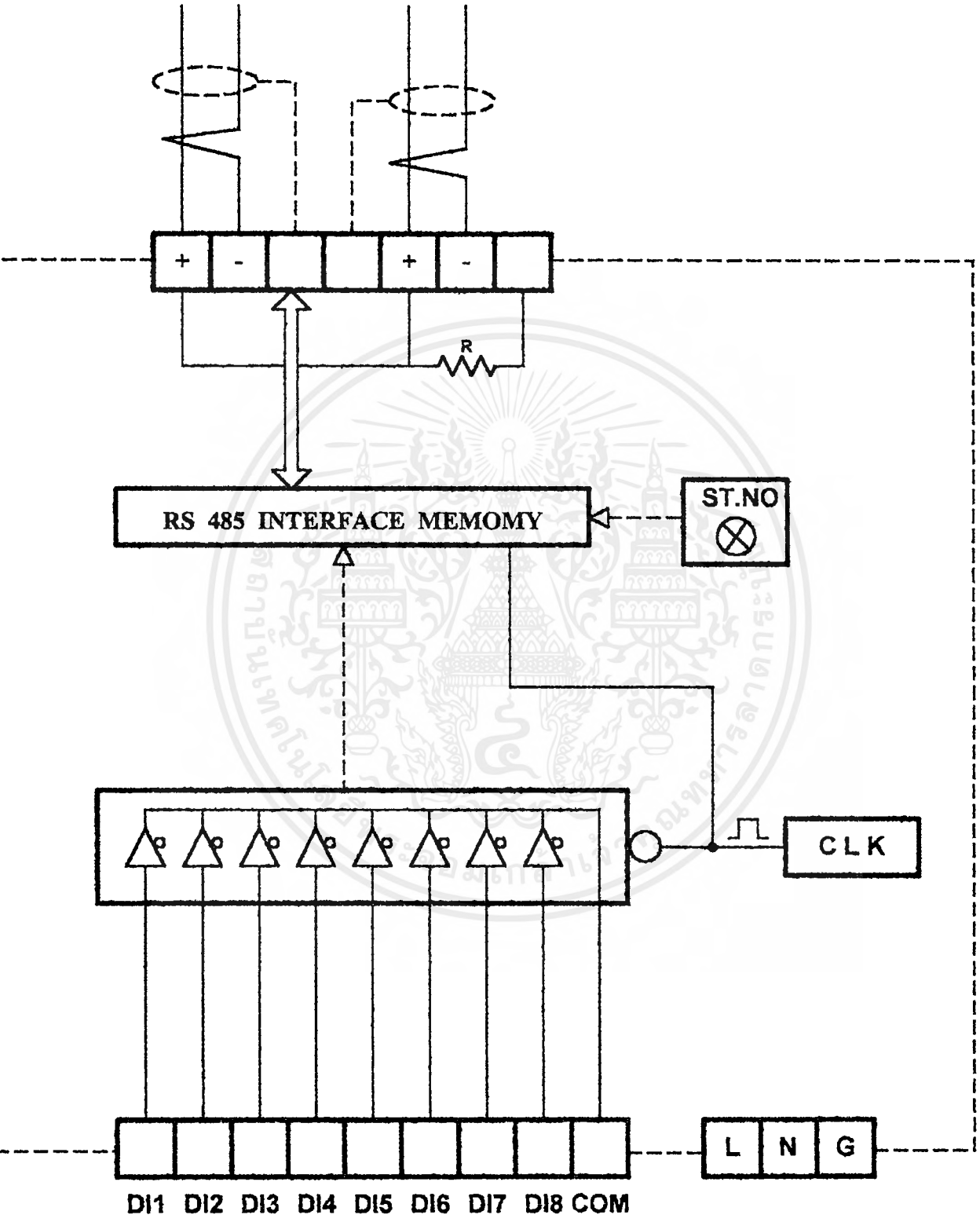
คุณสมบัติของ Main Board

CPU	80C31 (40 PIN-DIP OF MCS-51)
CLOCK	11.0592MHz
MEMORY	0/32K SOCKET (PROGRAM 0R DATA SELECTABLE)
PORT	8 BIT (PORT1 OF MCS-51)
	4 BIT (/INT0,/INT1,/T0,/T1 OF MCS-51)
	1 SERIAL PORT (RS232 OR RS485 SELECTABLE)
LED	1 POWER LED
SWITCH	1 RESET SWITCH
CONNECTOR	16 PIN PORT1 BUS (WITH /INT0,/INT1,/T0,/T1)
	40 PIN MCS-51 SYSTEM-BUS
	3 PIN RS232
	2 PIN RS485
	2 PIN 5V DC
JUMPER	2 WAY JUMPER FOR /EA SELECT (EXT,INT)
	2 WAY X 4 JUMPER FOR MEMORY SOCKET (ROM,RAM)
	2 WAY X 2 JUMPER FOR MEMORY SOCKET (8-16K,32K)
POWER SUPPLY	5V DC CURRENT 32 mA (WITH 27C256 EPROM)
PCB SIZE	8.8 X 7.1 cm

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DIGITAL INPUT UNIT

DATA Communication



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

9.6 หน้าที่หลักของ Digital Input Unit

Digital Input Unit เป็นหนึ่งใน Small Scaie DCS ซึ่งมีหน้าที่รับข้อมูลแบบ Digital จากภายนอก การรับข้อจากภายนอกจะเป็นข้อมูลขนาด 8 บิทแบบขนาน และจะทำการพักข้อมูลไว้ที่ Buffer ภายในและจะทำการ Latch ค่าข้อมูลไว้เพื่อทำการตรวจสอบข้อมูลที่ส่ง จากนั้นจะทำการแปลงข้อมูลขนาด 8 บิท ให้เป็นข้อมูลที่จะส่งออกแบบอนุกรมครั้งละ 1 บิท และข้อมูลที่ทำการแปลงเรียบร้อยแล้วจะทำการแปลงสัญญาณให้เป็นระบบมาตราฐาน RS-485 ซึ่งข้อมูลที่ได้นี้จะมีการทำส่งไปยังคอมพิวเตอร์ ซึ่งคอมพิวเตอร์จะมีการส่งสัญญาณเรียกข้อมูลมายังตัว Digital Input Unit ซึ่งสัญญาณที่คอมพิวเตอร์ส่งมาจะมาสั่งให้ CPU ในตัว Digital Input Unit เริ่มส่งข้อมูลที่แปลงแล้วไปยังคอมพิวเตอร์ โดยข้อมูลที่ส่งไปให้คอมพิวเตอร์จะถูกส่งผ่านไปที่ Interface Data Linker ซึ่งจะถูกแปลงสัญญาณที่ได้รับจาก Digital Input Unit เป็นสัญญาณในระบบมาตราฐาน RS-232 เพื่อที่จะติดต่อกับคอมพิวเตอร์อีกทีหนึ่ง ซึ่งจะเป็นการจบหน้าที่ของ Digital Input Unit

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

<u>หน้าที่ของ บล็อกต่าง ๆ</u>

บล็อกที่ 1 มีหน้าที่ รับข้อมูลแบบ Digital ขนาด 8 บิทแบบขนาน จากภายนอก

บล็อกที่ 2 มีหน้าที่ เป็น Buffer จะรับข้อมูลแบบ Digital ขนาด 8 บิทแบบขนานมาพักไว้ภายในชั่วคราวเพื่อที่จะส่งไปให้ CPU ทำการคำนวณและแปลงข้อมูลให้ได้ตามที่ต้องการ และจะทำการ Latch ค่าข้อมูลไว้เพื่อที่จะทำการตรวจสอบได้

บล็อกที่ 3 มีหน้าที่ เป็น Main Board ที่จะใช้เป็นตัวคำนวณและแปลงค่าข้อมูลแบบ8 บิท แบบขนานแล้ว จะส่งข้อมูลแบบอนุกรมออกไปครั้งละ 1 บิทโดยจะใช้ Microcontroller เป็นตัวแปลงข้อมูล และจะใช้ Microcontroller เป็นตัวที่จะติดต่อสื่อสารกับคอมพิวเตอร์อีกด้วย

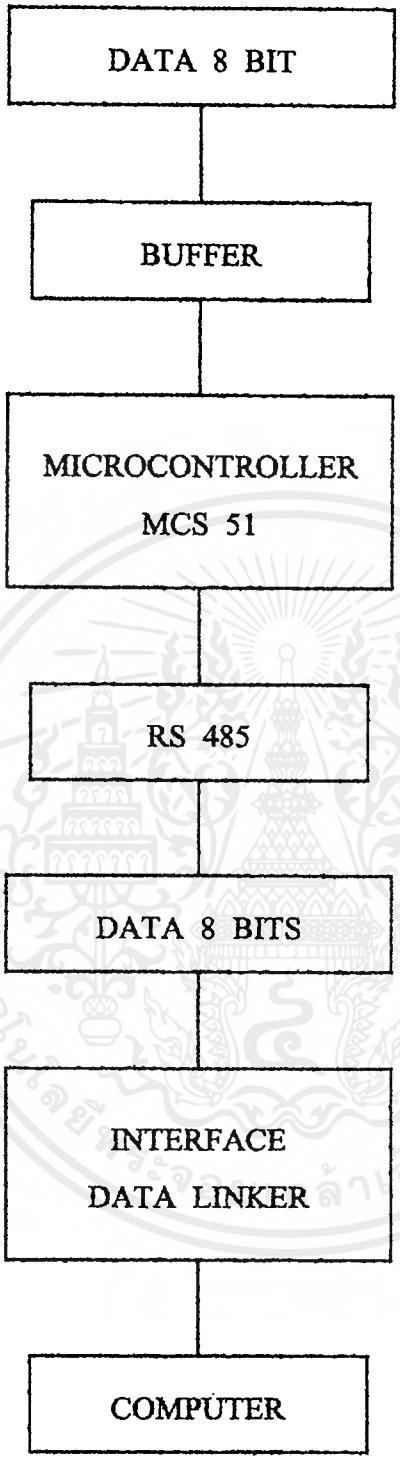
บล็อกที่ 4 มีหน้าที่ เป็นตัวที่จะรับข้อมูลที่ Microcontroller ส่งออกมาแล้วจะทำการแปลงสัญญาณรับมาให้เป็นระบบมาตราฐาน RS-485

บล็อกที่ 3 มีหน้าที่ เป็นบัฟเฟอร์ขนาด 8 บิท สำหรับการสื่อสารแบบอนุกรมทั้งการรับข้อมูลและการส่งข้อมูลโดยแบ่งเป็นบัฟเฟอร์ทางด้านการรับข้อมูล 1 ชุด 8 บิท และ บัฟเฟอร์ทางด้านการส่งข้อมูล 8 บิท โดยที่ CPU จะทำการจัดการเลือกบัฟเฟอร์โดยอัตโนมัติ

บล็อกที่ 6 มีหน้าที่ เป็นบล็อกของ Interface Data Linker จะทำการรับข้อมูลที่เป็นสัญญาณมาตราฐานRS-485 ให้เป็นสัญญาณในระบบมาตราฐาน RS-232

บล็อกที่ 7 มีหน้าที่ เป็นคอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



<u>บล๊อคไดอะแกรมการทำงานของ DIGITAL INPUT UNIT</u>

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

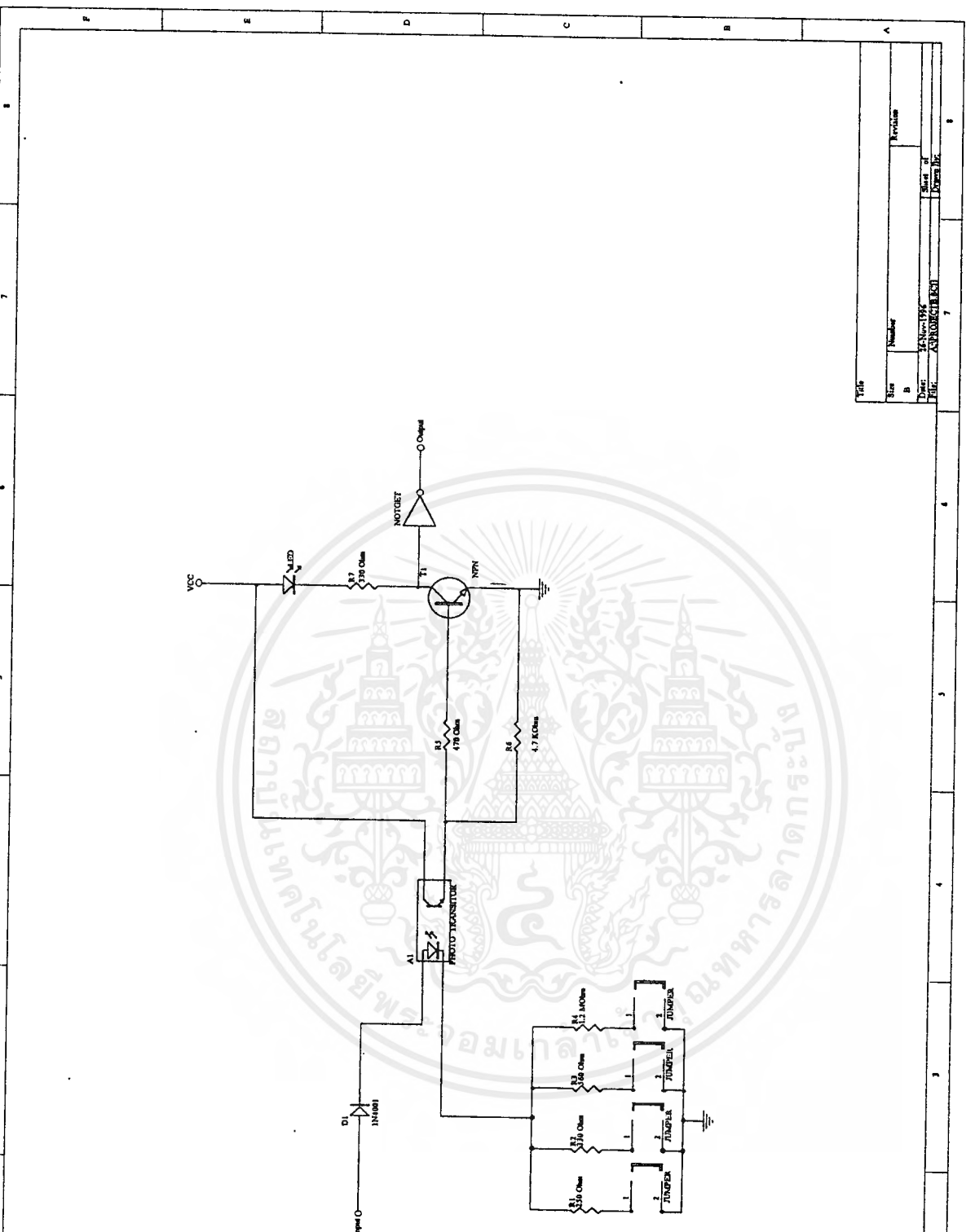
รายละเอียดของส่วนประกอบต่าง ๆ

1. ส่วนของ Photo Transitor

ทำหน้าที่รับสัญญาณและแสดงค่าของ สัญญาณ Input ที่รับเข้ามาจาก Process ซึ่งมีความสำคัญและประโยชน์ดังนี้ คือ จะเป็นส่วนที่เก็บค่า Input ที่เข้ามาใน ส่วนของ Digital Input Unit และจะเป็นตัวแสดงค่าของ Input ที่รับเข้ามาด้วย และยังมีประโยชน์มากคือเป็นตัวที่แยกสัญญาณ Imput จากภายนอกออกจากสัญญาณ Input ภายใน ซึ่งจะเป็นการป้องกันวงจรภายใน Digital Input Unit ด้วย หากเกิดการผิดพลาดจากวงจรภายนอก ซึ่งจะใช้ Photo Transitor เป็นตัวแยกและรับสัญญาณ Input แทนที่จะรับสัญญาณ Input จากภายนอกโดยตรง ซึ่งเราสามารถที่จะรับแรงดันได้หลายระดับโดยการใช้ค่าความต้านทานเป็นตัวป้องกันกระแสโดยสามารถที่เป็นค่าความต้านทานได้หลายค่าโดยการเปลี่ยนจัมเป้อร์ และในส่วนนี้ยังเป็นการแยกบิตแต่ละบิตให้เป็นอิสระต่อกันด้วยซึ่งเป็นผลให้บิตแต่ละบิตไม่จำเป็นต้องรับแรงดันที่เท่ากันก็ได้โดยเราจะต้องเลือกจัมเป้อร์ที่ค่าความต้านทานให้ถูกต้องก็เป็นการใช้ได้แล้ว ซึ่งในวงจรนี้เราได้ใส่ค่าความต้านทานไว้ 4 ระดับจึงทำให้สามารถรับค่าแรงดันได้ 4 ระดับ ดังนี้ คือ 1. 5V 2. 9V 3. 12V 4. 24V ซึ่งเราสามารถที่จะประยุกต์ได้อีกโดยการเพิ่มค่าความต้านทานเข้าแล้วใส่จัมเป้อร์เพื่อที่จะสามารถเลือกระดับแรงดันได้มากขึ้นไปอีก

ในส่วนของ LED เราได้นำมาแสดงผลให้ตรงกับ Terminal ในแต่ละบิตด้วยเพื่อเป็นการง่ายในการดู และตรวจสอบว่าในแต่ละบิตมีผลของข้อมูลเป็นอย่างไรบ้าง ส่วนในจุดที่เป็น Output ได้ใส่ Notget เพื่อให้สัญญาณ Output ที่จะส่งไปให้ Micro Controller มีค่าตามความเป็นจริงกับที่รับสัญญาณ Input เข้ามา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



วงจรส่วน Photo Transistor

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. เมนบอร์ด Micro Controller

ในการออกแบบเมนบอร์ด Micro Controller ส่วนนี้จะเป็นส่วนที่แสดงว่า Micro Controller นี้ได้ใช้บอร์ดของ V-31 ของ บริษัท ศิลารีเสิร์ช จำกัด เป็นมาตราฐานอ้างอิง ซึ่งสามารถใช้งานได้ตามความต้องการ ซึ่งมีรายละเอียดและส่วนประกอบต่าง ดังนี้

1.ส่วน Power ในส่วนนี้จะเป็นส่วนที่แสดงว่า Micro Controller พร้อมที่จะทำงานดดยจะแสดงผลด้วยหลอด LED สีแดง

2.ส่วนของ Switch Reset ในส่วนนี้จะเป็นส่วนที่จะเป็นตัว Reset Micro Controller เมื่อ Micro Controller เกิดทำงานผิดพลาด กรือเราต้องการให้ Micro Controller ทำงานใหม่ก็สามรถที่จะกด Switch Reset ได้ ซึ่งจะมีอยู่ 2 ที่คือ ที่บอร์ดของ Micro Controller และที่ตังกล่องของ Module Digital Input Unit

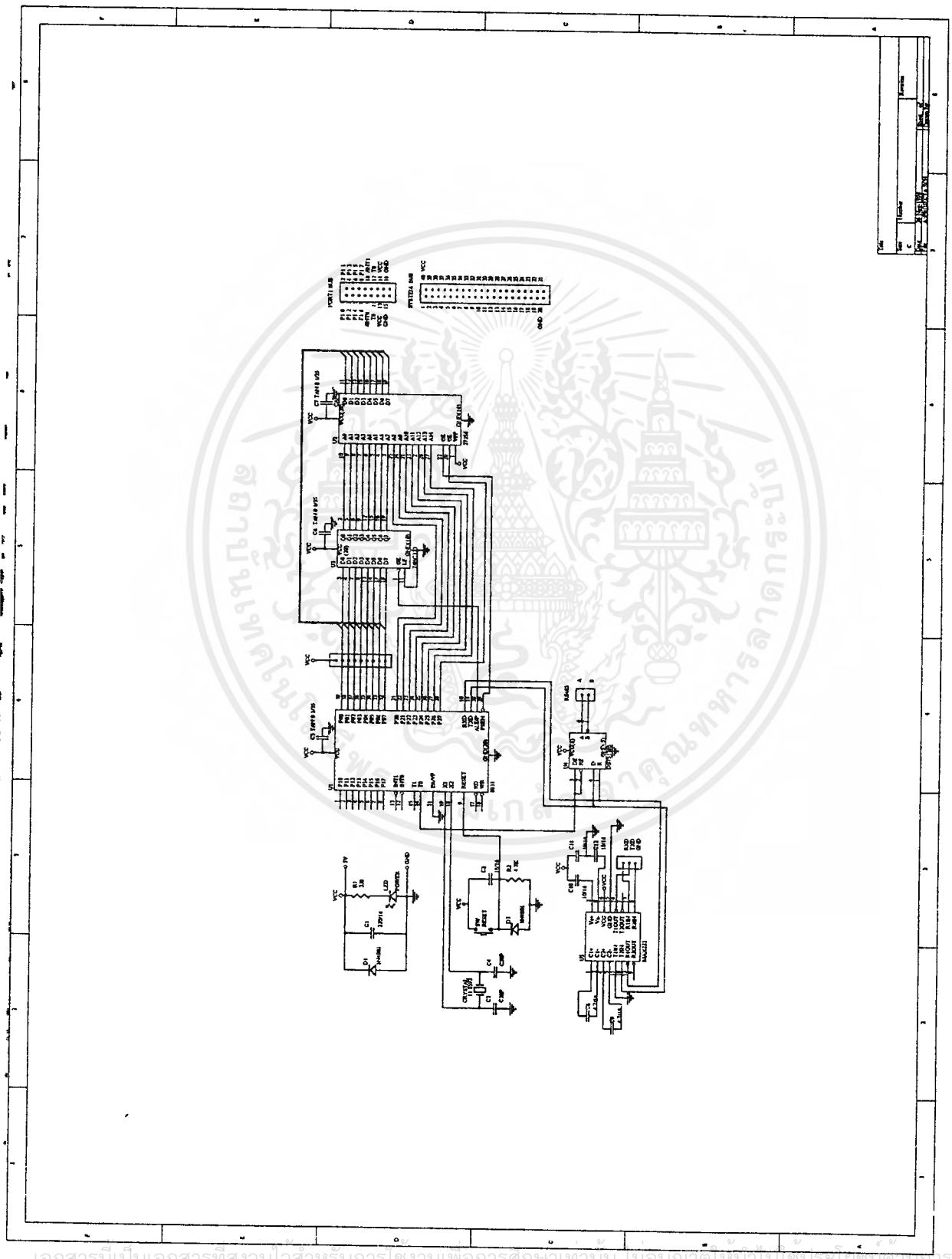
3.ส่วนสัญญาณนาฬิกา ในส่วนนี้จะใช้ Cystal ที่มีความถี่ 11.0592 MHz เป็นตัวสร้างสัญญาณนาฬิกาเพื่อป้อนให้กับ Micro Controller

4.ส่วนรับส่งข้อมูลอนุกรม ในส่วนนี้จเป็น Port ที่รับส่งข้อมูลแบบอนุกรมโดยได้ทำการต่อ Port ให้สามารถใช้งานรับส่งข้อมูลได้ 2 มาตราฐานคือ ในมาตราฐาน RS-485 และ มาตราฐาน RS-232 ซึ่งในมาตราฐาน RS-485 นี้จะใช้เพื่อติดต่อข้อมูลในระบบ RS -485 ที่ใช้ใน Project นี้ แต่ในระบบมาตราฐาน RS-232 ทำขึ้นเพื่อสำรองในการใช้ติดต่อกับ Computer ได้โดยตรง สำหรับในระบบมาตราฐาน RS-485 จะใช้ IC เบอร์ 75176 ในการเป็นตัวแปลงระดับสัญญาณเพื่อให้ Computer สามารถที่จะติดต่อกับ Micro Controller ได้ ส่วนในระบบมาตราฐาน RS-232 เราจะใช้ IC เบอร์ Max 232 เป็นตัวแปลงระดับสัญญาณเพื่อให้ Computer สามารถที่จะติดต่อกับ Micro Controller ได้

5.ส่วนของขา EA ของ Micro Controller ในส่วนนี้9ตามที่ได้ออกแบบไว้เราจะ Set ให้ขานี้ลงกราวด์ เพื่อที่จะทำให้ใช้ Program จากภายนอก ซึ่งเราจะใช้ Eprom เป็นตัวเก็บ Program และเมื่อเปิดเครื่องขึ้น Micro Controller จะทำการ Load ข้อมูลจาก Eprom เข้ามาเอง

6.ส่วนของ Port ขนาน ในส่วนนี้ จะใช้ Port 0 และ Port 2 เป็น Portที่จะติดต่อและกำหนดขนาดของ Eprom ซึ่งในบอร์ดนี้ได้กำหนดให้เป็น 32 Kbyte และจะใช้ IC เบอร์ 74HC373 เป็นสลับและแยกการส่ง ข้อมูลและแอดเดรสของ Eprom และ Port 1 จะใช้ในการรับ Input ที่ส่งเข้ามาจาก

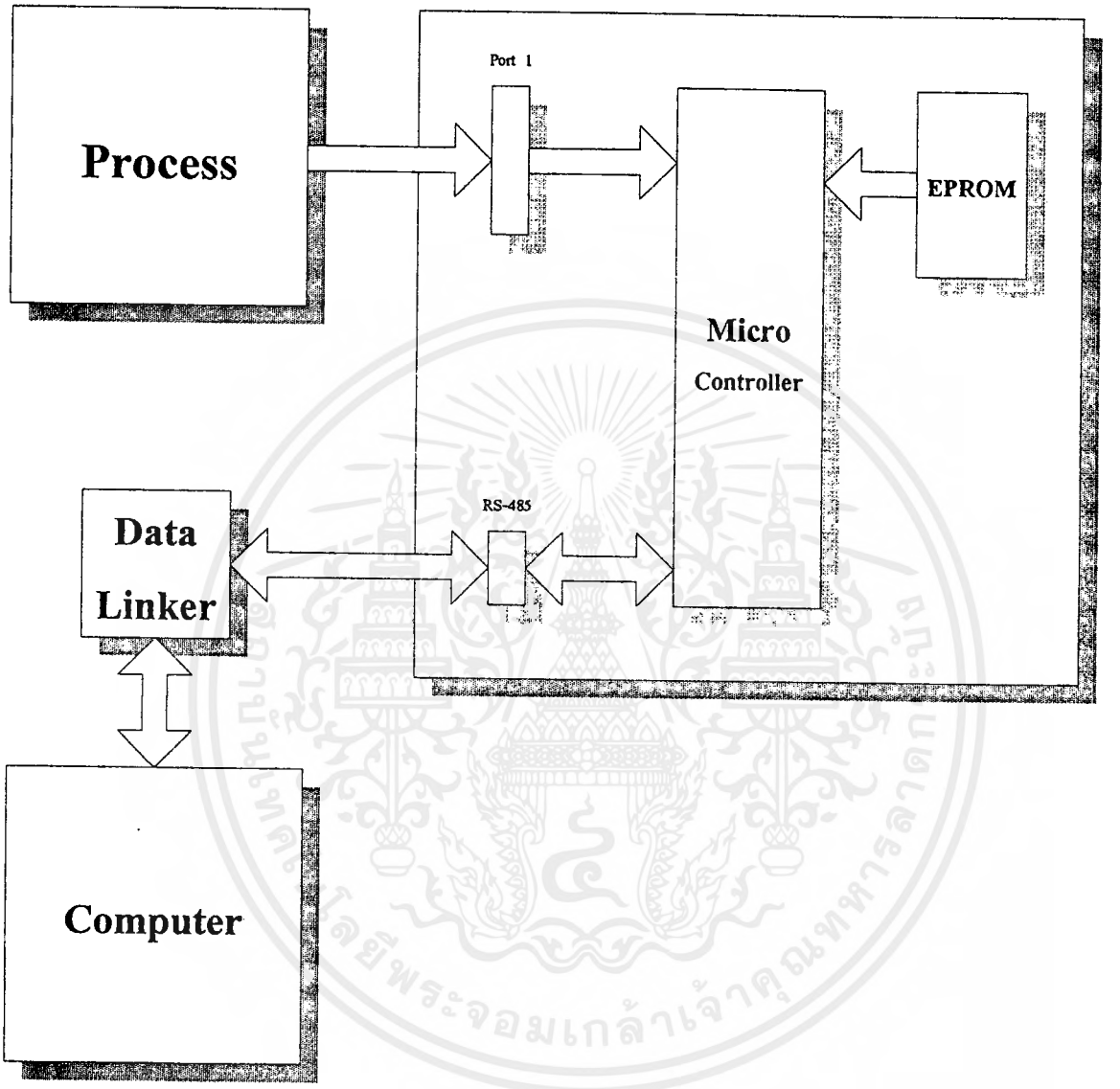
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



วงจรเมนบอร์ด Micro Controller

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Digital Input Unit



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภายนอก โดยผ่าน วงจรของ Photo Transitor แล้ว โดยจะรับเข้ามาครั้งละ 8 บิตซึ่งจะเป็น ณ ปัจจุบันนั้น

3. การทำงานของ Digital Input Unit

เริ่มจากเมื่อเปิดเครื่อง Digital Input Unit Micro Controller จะทำการ Load Program จาก Eprom เข้าสู่ตัว Micro Controller เพื่อพร้อมที่จะทำงาน ในขณะเดียวกับ ถ้ามีข้อมูล Input เข้ามาข้อมูลจะค้างอยู่ที่ Port Input ที่มีขนาด 8 บิตซึ่งไม่สามารถเข้ามาในตัว Micro Controller ได้เพราะยังไม่ได้เรียกใช้ข้อมูล จากนั้นจะต้องรอจนกวา Computer จะทำการส่งสัญญาณ Protocal เข้ามาซึ่งเราจะต้องกำหนดมาตราฐานของสัญญาณนี้ก่อนซึ่งได้มีการกำหนดไว้แล้วซึ่งสัญญาณนี้จะเข้ามาทุก Unit ที่มีอยู่ เพียงแต่ว่า Micro Controller ของแต่ละ Unit จะเป็นตัวที่จะเช็คเองว่า Computer เรียก ST No ถูกต้องหรือเปล่า ซึ่งสัญญาณ Protocal ที่เข้ามาเป็นสัญญาณของ St No นี้หรือเปล่า ใน Unit นี้ได้กำหนดให้เป็น ST No 5 Micro Controller จะทำการเช็คว่าสัญญาณที่ Computer ส่งเข้ามาเป็น ST No 5 หรือเปล่า ถ้าใช่ ซึ่งแสดงว่าตรงกับ Module Digital Input Unit Micro Controller จะทำการนำข้อมูลที่รับมาจาก Port Input ขนาด 8 บิต เข้ามา แล้วทำการแปลงสัญญาณที่เป็นเลขฐาน 2 ให้เป็นเลขฐาน 16 ขนาด 2 บิต แล้วนำไปใส่ไว้ใน สัญญาณ Protocal ที่เรากำหนดไว้แล้ว จากนั้น Micro Controller จะทำการส่งสัญญาณ Protocal ที่มีข้อมูลเรียบร้อยแล้วกลับคืนสู่ Computer ทาง Port อนุกรม ซึ่งจะเป็นการเสร็จสิ้นขั้นตอนการทำงานของ Digital Input Unit

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การนำ Module Input Digital ไปใช้งาน

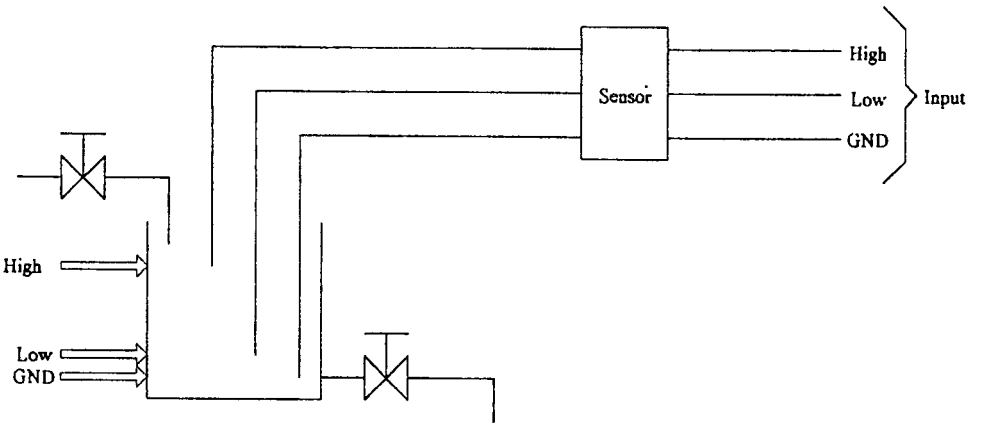
Module Input Digital นี้มี Input อยู่ด้วยกัน 8 บิท(bit)โดยแต่ละบิท(bit)เป็นอิสระต่อกัน กล่าวคือเราสามารถนำสัญญาน Output จากอุปกรณ์อื่นที่เป็นสัญญาน Digital หรือสัญญาน On-Off มาต่อเข้า Module Input Digital ได้จำนวน 8 ช่องสัญญานด้วยกันและอุปกรณ์ที่ให้สัญญาน Digital หรือสัญญาน On-Off ในงานอุตสาหกรรมมีพวก

อุปกรณ์ตรวจจับ(Sensor) อุปกรณ์เปิดปิดแบบไฟฟ้าแบบต่างๆ เป็นต้นและสัญญานที่นิยนำมาใช้ก็เป็นพวกอุปกรณ์ตรวจจับ(Sensor)เช่น อุปกรณ์ตรวจวัดระดับน้ำ อุปกรณ์ตรวจวัดอุณหภูมิ เป็นต้นโดยอุปกรณ์พวกนี้จะมีสัญญานส่วนใหญ่ออกมาสองสัญญานคือสัญญานที่มีค่าสูงกว่า(High)ค่าที่เราตั้งไว้และสัญญานที่มีค่าต่ำกว่า(Low)ค่าที่เราตั้งไว้โดยเราจะนำสัญญานที่ส่งออกมานำไปต่อเข้า Input ของ Module Input Digital โดยเราแยกแต่ละสัญญานต่อหนึ่ง Input ซึ่งถ้าอุปกรณ์ตรวจจับ(Sensor)ส่งสัญญานออกมาเพียงตัวละสองสัญญานก็สามารถจะต่ออุปกรณ์ตรวจจับ(Sensor)ได้ Module ละสี่อุปกรณ์ โดยสัญญานที่ส่งเข้ามาที่ Module Input Digital จะถูกนำไปเก็บไว้ที่ Buffer ก่อนเพื่อรอการเรียกจากคอมพิวเตอร์(Computer)เพื่อนำไปประมวลผลต่อไปซึ่งการประมวลผลของคมพิวเตอร์ขึ้นอยู่กับโปรแกรมทเขียนไว้ในคอมพิวเตอร์

ตัวอย่างการทดลอง

ในที่นี้เราใช้อุปกรณ์ตรวจจับแบบวัดระดับน้ำโดยมีการวัด 2 ระดับ คือ High และ Low ซึ่งเมื่อระดับน้ำที่อยู่ในถังถึงระดับ High จะทำให้อุปกรณ์ตรวจจับส่งสัญญาณ On ออกไป ซึ่งสัญญาณนี้เราสามารถนำไปเป็น สัญญาณ Input ให้กับ Digital Input Unit ส่วนในระดับ Low สัญญาณที่ได้จากตัวตรวจจับจะ On อยู่ตลอดเวลา เมื่อระดับน้ำลดลงเกินระดับ Low จะทำให้สัญญาณที่ออกจากตัวตรวจจับ Off สัญญาณที่จะส่งไปให้ Digital Input Unit ซึ่งจะมีรูปการต่อใช้งานดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปตัวอย่างการใช้งานวัดระดับน้ำ

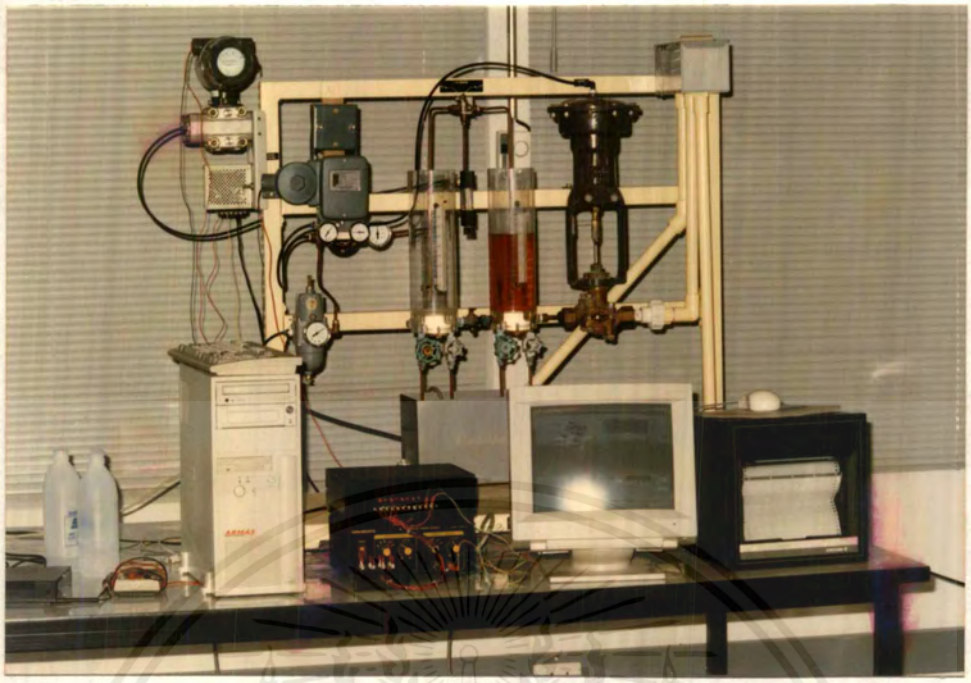
ปัญหาและอุปสรรค

ในการทดลอง Digital Input Unit ปัญหาที่พบแบ่งออกได้ดังนี้

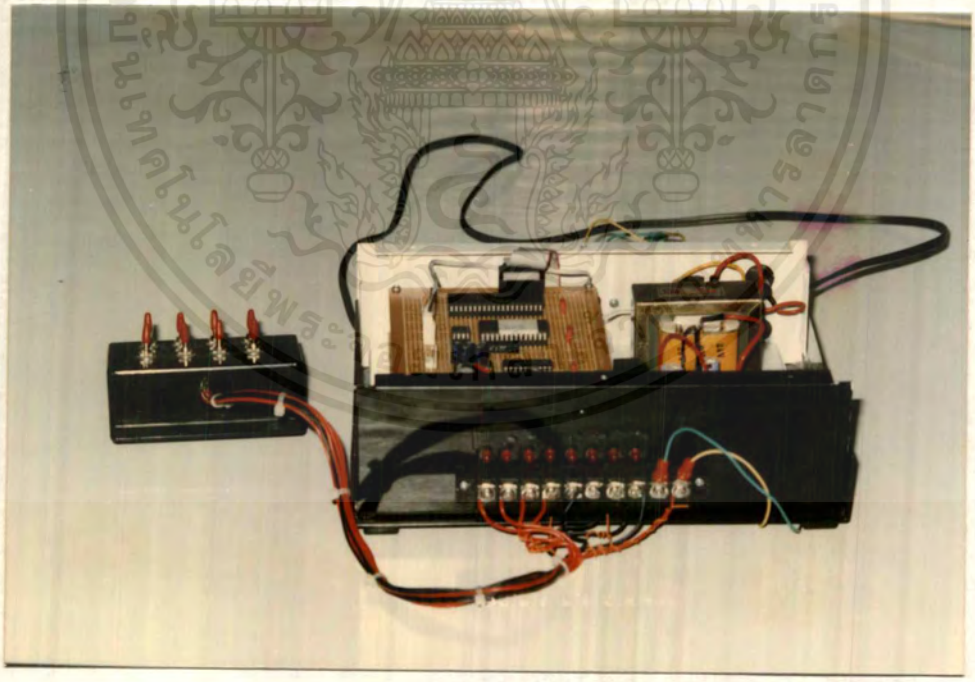
1.อุปกรณ์ที่ใช้ในวงจรที่ได้ออกแบบในแต่ละวงจรจะพบปัญหาว่าอุปกรณ์ที่ใช้ไม่ได้คุณภาพเมื่อมีการเปลี่ยนอุปกรณ์ที่มีคุณภาพที่ดีขึ้นวงจรจึงทำงานได้ตามที่ได้ออกแบบไว้

2.ในการทำบอร์ด Micro Controller ที่ใช้ เมื่อมีการออกแบบลายปรินซ์เรียบร้อยแล้ว ทำการลงอุปกรณ์ ปรากฏว่า มีปัญหาในส่วนของ Eprom และ CPU จึงได้ใช้ การแก้ปัญหา โดยการลงอุปกรณ์บนแผ่นปรินซ์อเนกประสงค์และทำการวายสายตามวงจรที่ได้ออกแบบไว้ จึงสามารถแก้ปัญหาได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

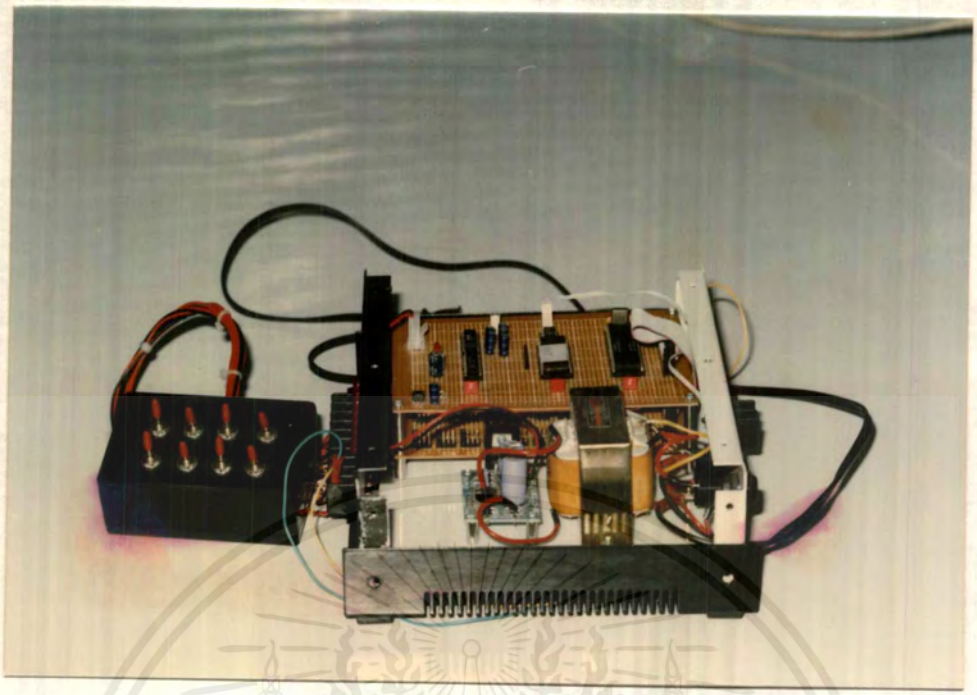


รูปแสดง UNIT 05 (DIGITAL INPUT UNIT) กับการควบคุมกระบวนการระดับน้ำ

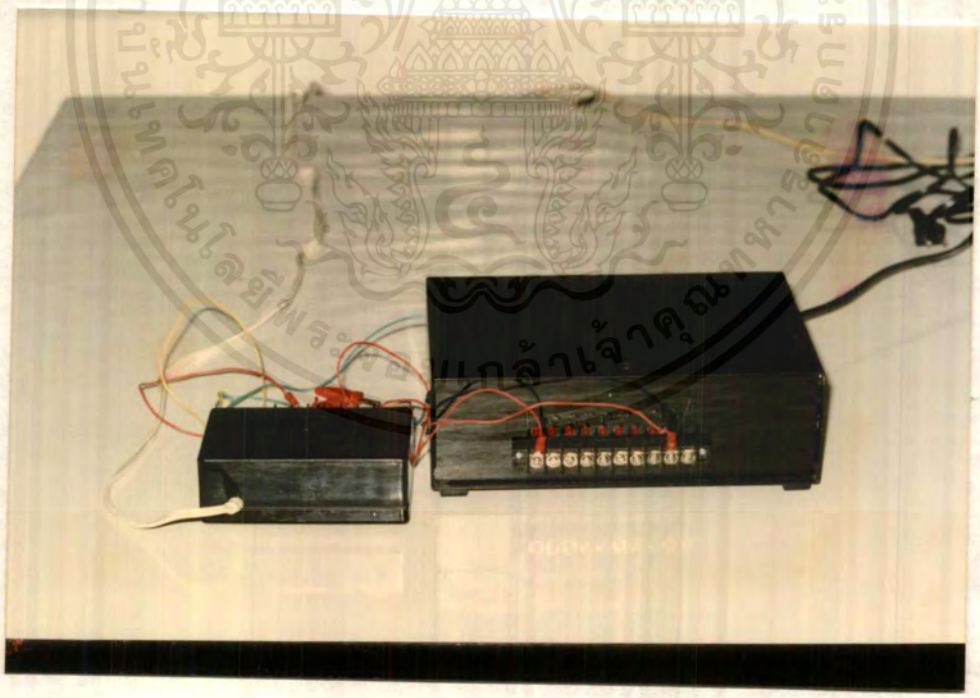


รูปแสดง UNIT 05 (DIGITAL INPUT UNIT)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

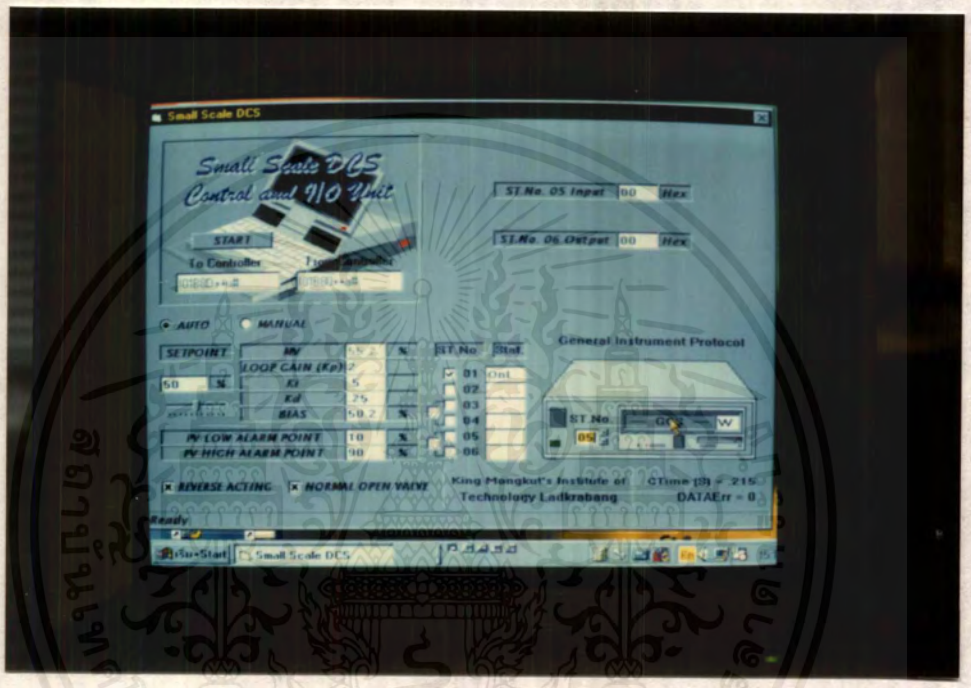


รูปแสดงการต่อวงจรภายในของ UNIT 05



รูปแสดง UNIT 05 กับชุด LEVEL SENSOR

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

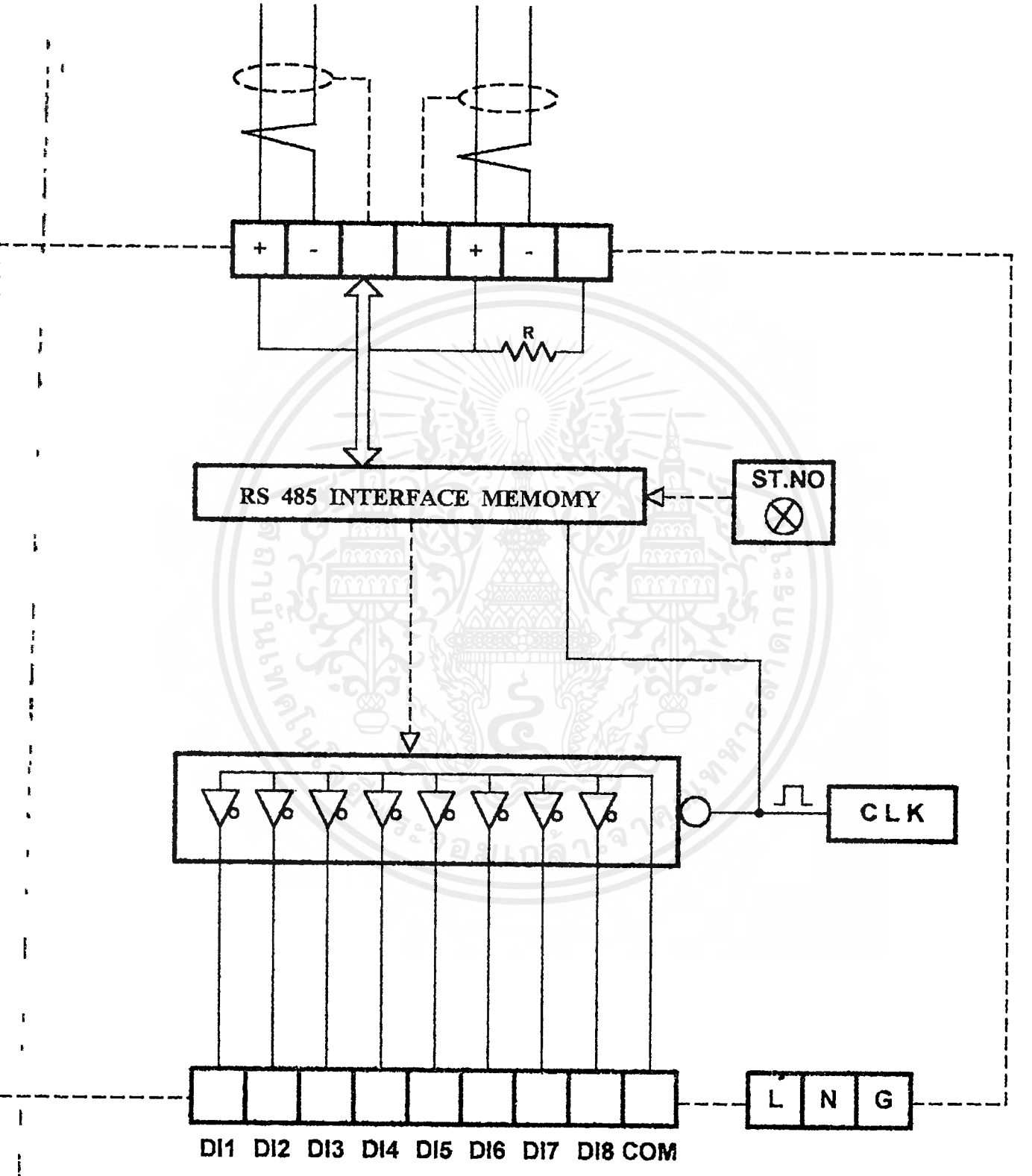


รูปแสดงหน้าจอของ ST.No. 05

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

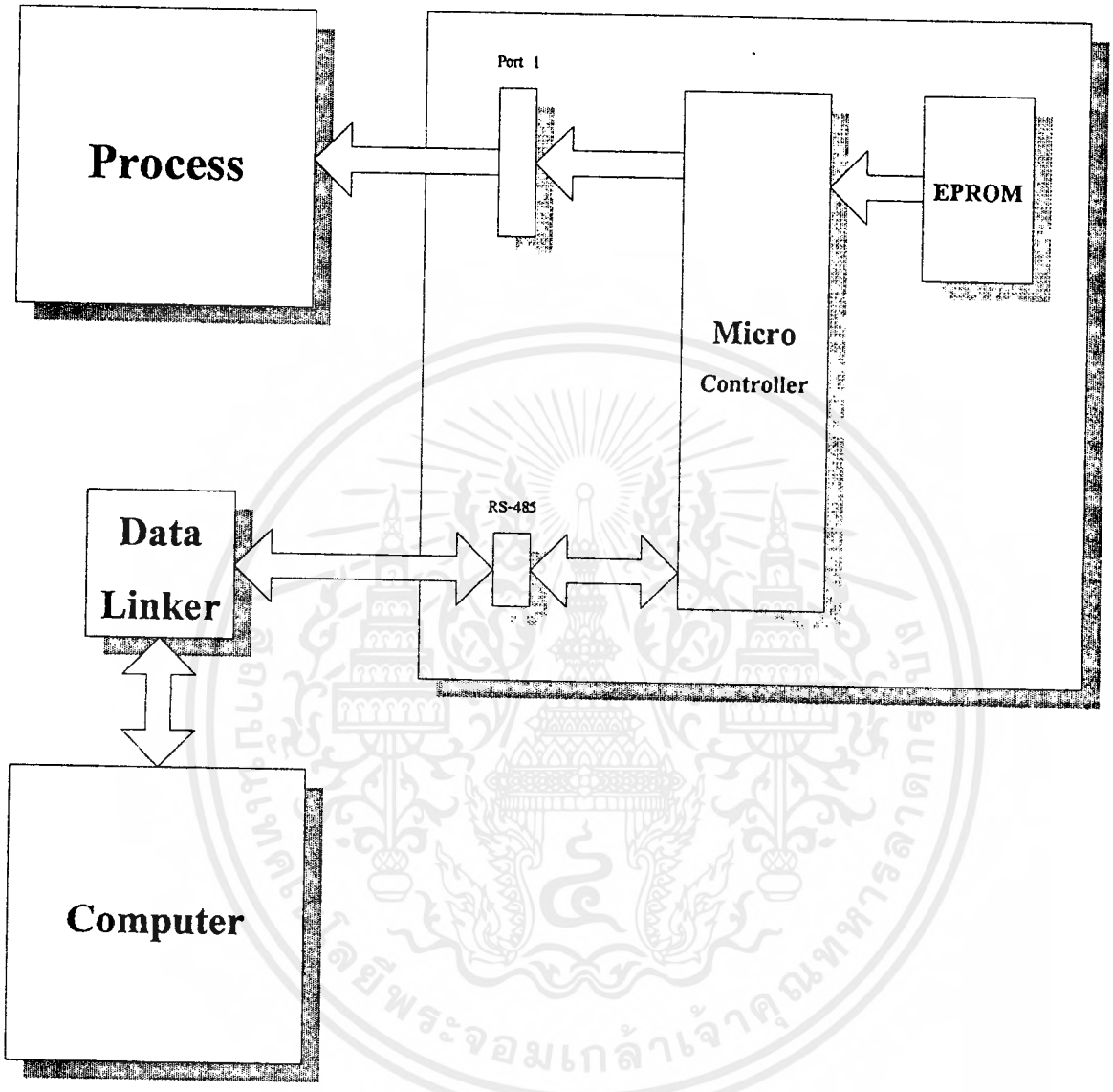
DIGITAL OUTPUT UNIT

DATA Communication



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Digital Output Unit



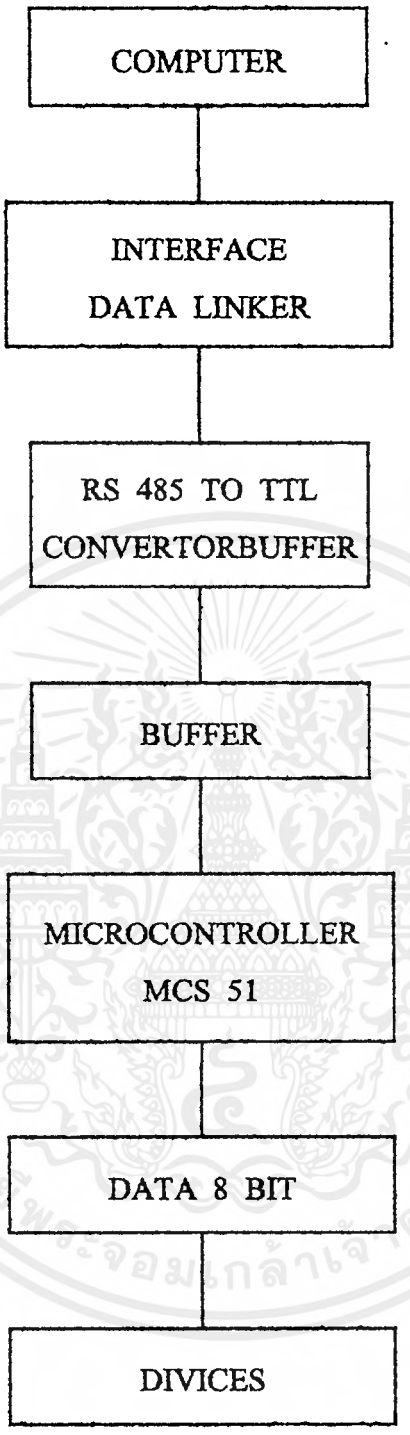
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

9.7 หน้าที่หลักของ Digital Output Unit

ภาค DIGITAL OUTPUT UNIT เป็นส่วนหนึ่งในหลาย ๆ ส่วนของระบบ SMALL SCALE DCS โดยในส่วนของ DIGITAL OUTPUT UNIT มีลักษณะการทำงานดังนี้

DIGITAL OUTPUT UNIT จะได้รับข้อมูลจากสายสัญญาณเป็นการส่งในระดับ RS 485 ซึ่งลักษณะของข้อมูลที่ส่งมาตามสายสัญญาณจะเป็นข้อมูลแบบอนุกรม (SERIES DATA) เมื่อมีสัญญาณข้อมูลเข้ามา ระดับสัญญาณ RS 485 จะถูกแปลงระดับสัญญาณลงให้อยู่ระดับสัญญาณแบบ DIGITAL โดยใช้ IC # 75176 เป็นตัวแปลงระดับสัญญาณลง จากนั้นข้อมูลก็จะถูกนำเข้าไปเก็บใน SERIES BUFFER ของ MICROCONTROLLER แล้ว MICROCONTROLLER จะทำการตรวจข้อมูลชุดที่ส่งมาให้นั้นว่ามีตำแหน่ง (ADDRESS) ตรงกับ DIGITAL OUTPUT UNIT หรือไม่ เมื่อตรวจสอบแล้วพบว่า ตำแหน่ง (ADDRESS) ตรงกับส่วนของ DIGITAL OUTPUT UNIT ก็จะทำการรับข้อมูลนั้นเข้ามา แล้วทำการแปลงข้อมูลให้เป็นข้อมูลแบบขนานขนาด ๘ บิท ส่งออกไปทาง PORT 1 ต่อไป แต่ถ้าข้อมูลชุดที่ส่งมานั้นมีตำแหน่ง (ADDRESS) ไม่ตรงกับ DIGITAL OUTPUT UNIT แล้ว MICROCONTROLLER จะไม่กระทำการใด ๆ กับข้อมูลชุดนี้อีก และที่ PORT 1 ค่าข้อมูลที่ปรากฏอยู่ก็จะยังคงเป็นค่าของข้อมูลเดิม ก่อนหน้าที่จะมีข้อมูลส่งมายังส่วน DIGITAL OUTPUT UNIT แล้วถูก MICROCONTROLERปฏิเสธข้อมูลชุดนั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



บล๊อคไดอะแกรมการทำงานของ DIGITAL OUTPUT UNIT

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

<u>หน้าที่ บล็อกต่าง ๆ</u>

บล็อกที่ 1 มีหน้าที่ เป็นคอมพิวเตอร์

บล็อกที่ 2 มีหน้าที่ เป็นบล็อกของ Interface Data Linker จะทำหน้าที่รับข้อมูลที่เป็นสัญญาณมาตราฐาน RS-232 ให้เป็นสัญญาณในระบบมาตราฐาน RS-485

บล็อกที่ 3 มีหน้าที่ เป็นตัวที่จะรับข้อมูลที่คอมพิวเตอร์ส่งออกมาแล้วทำการที่ได้รับในระบบ RS-485 ให้เป็นระดับสัญญาณ TTL แล้วส่งให้ Micricontroller

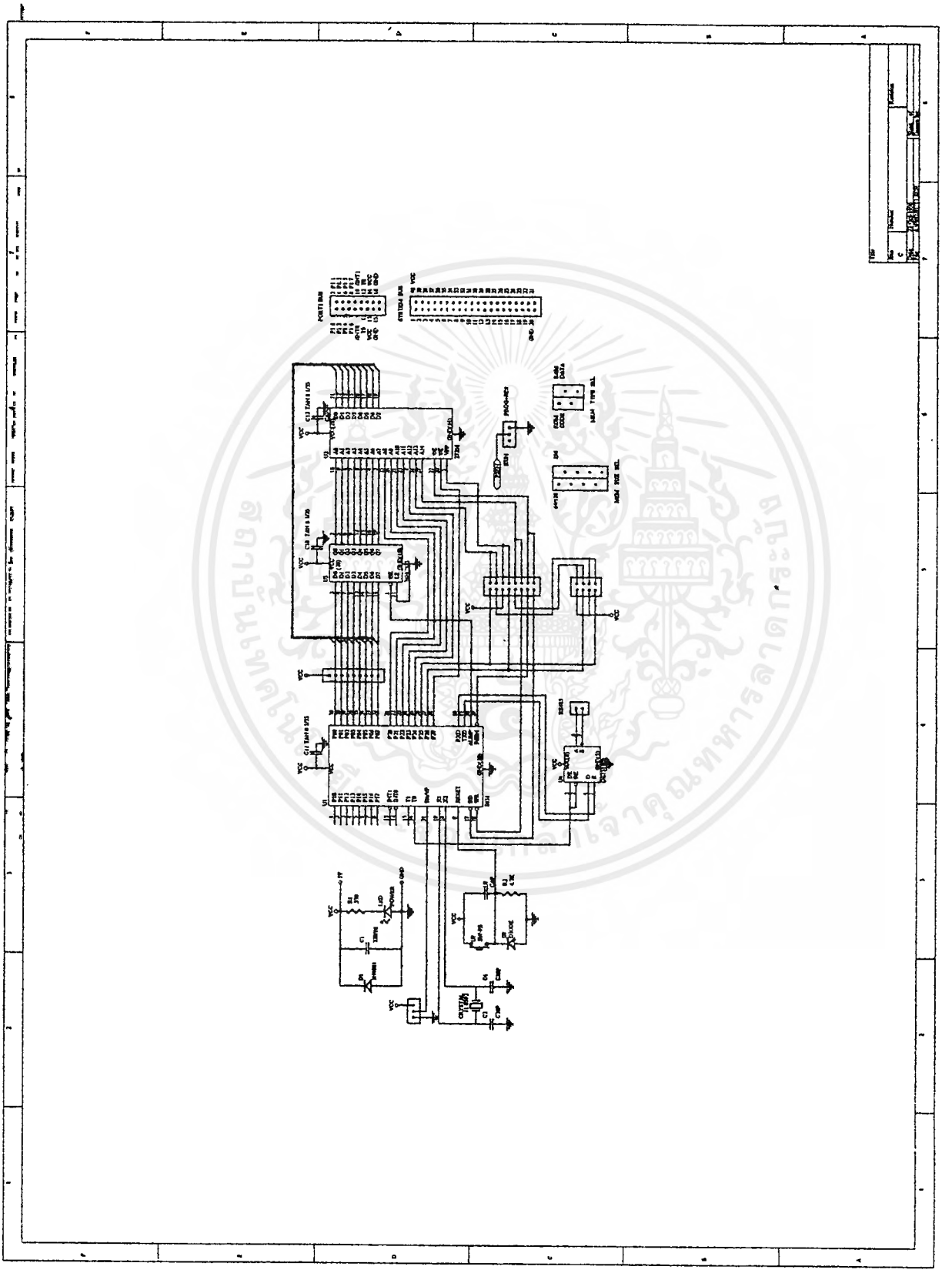
บล็อกที่ 4 มีหน้าที่ เป็นบัฟเฟอร์ขนาด 8 บิท สำหรับการสื่อสารแบบอนุกรมทั้งการรับข้อมูลและการส่งข้อมูลโดยแบ่งเป็นบัฟเฟอร์ทางด้านการรับข้อมูล 1 ชุด 8 บิท และ บัฟเฟอร์ทางด้านการส่งข้อมูล 8 บิท โดยที่ CPU จะทำการจัดการเลือกบัฟเฟอร์โดยอัตโนมัติ

บล็อกที่ 5 มีหน้าที่ เป็น Main Board ที่จะใช้เป็นตัวคำนวณและแปลงค่าข้อมูลแบบ 1 บิท แบบอนุกรม แล้วจะส่งข้อมูลแบบขนานออกไปครั้งละ 8 บิทโดยจะใช้ Microcontroller เป็นตัวแปลงข้อมูล และจะใช้ Microcontroller เป็นตัวที่จะติดต่อสื่อสารกับคอมพิวเตอร์อีกด้วย

บล็อกที่ 6 มีหน้าที่ รับข้อมูลขนาด 8 บิทแบบขนานจาก Micricontroller แลจะทำการส่งไปภายนอก

บล็อกที่ 7 มีหน้าที่ เป็นอุปกรณ์ภายนอก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



วงจรเมนบอร์ด Micro Controller v-31

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทดสอบ DIGITAL OUTPUT 8 BIT

การประยุกต์ใช้งาน digital output สามารถใช้งานในรูปแบบต่างๆเช่นใช้เป็นส่วนบันทึกและแสดงผล

ขับอุปกรณ์หรือใช้เป็นส่วนเตือนภัยเป็นต้น

การทดสอบdigital output ทดสอบโดยการสร้างชุดขับสเต็ปมอเตอร์ซึ่งเป็นขนาด4เฟสแบบยูนิโพล่า การทดสอบจึงใช้งานเพียงครั้งละ4บิตโดยกำหนดบิตทดสอบที่ส่วนของโปรแกรมหรือส่วนฮาร์ดแวร์โดยที่ชุดขับสเต็ปมอเตอร์ จะมีวงจรแยกกราวด์เพื่อเป็นการป้องกันแรงดันป้อนกลับโดยใช้ตัวเชื่อมโยงทางแสง (ออบโตคัปเปอร์) เบอร์ 4N26 ภายในเป็นโฟโตทรานซิสเตอร์ พื้นฐานการทำงานของตัวเชื่อมโยงทางแสงมีดังนี้

ตัวเชื่อม ต่อ ผ่านแสง

(optocoupler devices)

ตัวเชื่อมต่อผ่านแสง (optocoupler) หรือตัวแยกวงจรด้วยแสง (optoisolator) เป็นอุปกรณ์ที่ใช้ในเชื่อมต่อวงจรสองส่วนที่มีการส่งผ่านข้อมูลระหว่างกัน ที่จำเป็นต้องมีการแยกกันของวงจรทั้งสองส่วนนี้อย่างเด็ดขาด เช่น ใช้ในวงจรเปลี่ยนค่าระดับลอจิกระหว่างวงจรใช้กัน สัญญาณรบกวนที่จะส่งผ่านจากวงจรหนึ่งไปยังวงจรอื่น แยกระดับลอจิกของวงจรออกจากแรงดันไฟบ้าน และใช้ในการแก้ปัญหาวงรอบของดิน (ground loop)

การใช้งานตัวเชื่อมต่อผ่านแสง จะสามารถใช้ในการเชื่อมต่อสัญญาณที่เป็นกระแสตรงได้ดี โดยที่ยังคงความสามารถในการแยกวงจรอินพุทออกจากกันทางไฟฟ้าได้ ในบางกรณีก็จะนำไปใช้แทนรีเลย์หรือหม้อแปลงของวงจรดิจิตอลได้

พื้นฐานของตัวเชื่อมต่อผ่านแสง

โครงสร้างของตัวเชื่อมต่อผ่านแสงจะประกอบด้วยตัวกำเนิดแสงที่เป็น LED เปล่งแสงอินฟาเรด (ปกติจะสร้างจากสารแกลเลี่ยมอาร์เซไนด์) เพื่อส่งแสงไปยังตัวรับแสงแบบซิลิคอน (silicon photodetector) ที่เป็นได้ทั้ง โฟโตทรานซิสเตอร์ โฟโตไดโอด หรือ อุปกรณ์ไวแสงอื่น ๆ) ที่ทั้งหมดนี้ประกอบอยู่ในตัวถังแบบทึบแสงเพื่อป้องกันแสงจากภายนอกมารบกวน ดังที่แสดงได้ในรูปที่ 1 ที่เป็นภาพตัดขวางแสดงภายในของตัวเชื่อมต่อผ่านแสงแบบ 1 แชนแนล ในตัวถังแบบ DIP 6 ขา อินฟาเรด LED หรือ IRED ที่ใช้เมื่อมีกระแสไฟไหลผ่าน (ไบแอสตรง) จะเปล่งแสงอินฟาเรดในช่วงความยาวคลื่น 900-940 นาโนเมตร (nm) ตัวรับแสงจะเป็นโฟโตทรานซิสเตอร์แบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

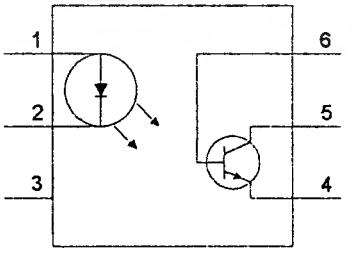
NPN ที่ไวต่อแสงความยาวคลื่น 900-940 นาโนเมตร (nm) เช่นกัน ทั้งตัว IRED และโฟโตไดโอดจะถูกสร้างบนไดซ์หรือชิบ (die หรือ chip) ตัวเดียวกัน

หลังจากที่ได้แผ่นชิบของอุปกรณ์แล้วก็จะนำมาประกอบบน leadframe ที่สร้างจากการนำแผ่นโลหะบางมาปั๊มพ์ให้เป็นรูปแบบต่าง ๆ โดยส่วนของ leadframe ที่ต่อภายในจะเป็นฐานรองของตัวชิบ ส่วนที่อยู่ภายนอกก็จะเป็นขาต่อของอุปกรณ์ที่ในกรณีนี้ก็จะเป็นขาตัวถังแบบ DIP

หลังจากที่ได้เดินสายต่อจากตัวชิบไปยังขาต่อ (leadframe) จะเทเรซินโปร่งแสงอินฟาเรดรอบ ๆ ตัวอุปกรณ์บนชิบทั้งสอง เพื่อให้เป็นช่องให้แสงผ่านหรือ ท่อนำคลื่นแสง (optical wavequide) แล้วนำอุปกรณ์ที่ได้ไปหล่อด้วยอีปอกซี่ทึบแสงให้เป็นรูปตัวถังแบบ DIP แล้วจึงงอส่วนของขาต่อลงให้เป็นตัวถังแบบ DIP ที่สมบูรณ์แบบ

รูปที่ 2 เป็นไดอะแกรมของขาต่อ (pin diagram) ของตัวเชื่อมต่อผ่านแสงแบบโฟโตทรานซิสเตอร์ตัวถังแบบ DIP 1 แชนแนลเป็นแบบยอดนิยม สาเหตุที่เรียกอุปกรณ์พวกนี้ว่าเป็น "ตัวเชื่อมต่อผ่านแสง" เนื่องจากพลังงานที่ส่งผ่านจากอินพุท (IRED) ไปยังเอาท์พุท (โฟโตทรานซิสเตอร์) จะเป็นแสงอินฟาเรด ส่วนที่เรียกว่าเป็น "อุปกรณ์แยกวงจรทางแสง" (optoisolator) เนื่องจากไม่มีกระแสไฟฟ้าส่งผ่านระหว่างวงจรทั้งสอง (ระหว่างตัวกำเนิดแสงและตัวตรวจจับแสงจะเป็นฉนวน) นอกจากนี้แล้วยังมีชื่ออื่น ๆ อีก เช่น Photocoupler หรือ Photon-coupled isolator

จากตัวถังแบบ DIP 6 ขา ของตัวเชื่อมต่อผ่านแสง ที่ขา 6 จะเป็นขาเบสของโฟโตทรานซิสเตอร์ ซึ่งตามการใช้งานปกติจะปล่อยว่างไว้ ส่วนขา 3 เป็นขาว่าง ในการใช้งานสามารถนำโฟโตทรานซิสเตอร์นี้มาใช้เป็นโฟโตไดโอดได้โดยการนำขาเบส (ขา6) มาต่อกับขาอีมิตเตอร์ (ขา 4) ซึ่งถ้าเป็นตัวเชื่อมต่อผ่านแสงที่เป็นตัวถังแบบ DIP 4 ขา หรือเป็นแบบหลายแชนแนลที่ไม่ได้ต่อขาเบสให้ ก็ไม่สามารถทำได้ ตามปกติแล้วตัวเชื่อมต่อผ่านแสงเอาท์พุทโฟโตไดโอดจะใช้งานกับงานที่ต้องการแบนด์วิดซ์กว้างและวงจรความเร็วสูง เช่น อุปกรณ์สื่อสารข้อมูล แต่ก็มีข้อเสียที่ประสิทธิภาพการเชื่อมต่อที่ต่ำมาก (โฟโต้ไดโอดจะมีประสิทธิภาพต่ำ)



รูปที่ 2 ผังวงจรของตัวเชื่อมต่อผ่านทางแสงเอ้าท์พุทโฟโตทรานซิสเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คุณสมบัติของตัวเชื่อมต่อผ่านแสง

ค่าคุณสมบัติที่สำคัญที่เป็นตัวกำหนดประสิทธิภาพการเชื่อมต่อผ่านแสงของตัวเชื่อมต่อผ่านแสงก็จะเป็นค่า อัตราส่วนการส่งผ่านกระแส (CTR ; Current Transfer Ratio) ที่เป็นอัตราส่วนของกระแสเอาท์พุทต่อกระแสอินพุทที่ค่ากระแสไบแอสที่กำหนดไว้ ที่มีสมการดังนี้

CTR = (Ice0)/(IF) x 100% หน่วยเป็นเปอร์เซ็นต์

ค่าอัตราส่วนการส่งผ่านกระแสจะมีค่าสูงสุดเมื่อแสงอินฟาเรดของ IRED มีความถี่ (หรือสเปกตรัม) ตรงกับตัวรับแสงที่เอ้าท์พุท

ตัวอย่างที่CTR 100% หมายความว่า เมื่อป้อนกระแสให้ IRED 1mA ก็จะได้กระแสเอ้าท์พุท 1mA ตัวเชื่อมต่อผ่านแสงแบบเอ้าท์พุทโฟโตทรานซิสเตอร์ดังที่แสดงในรูปที่ 1 และ 2 จะมีค่า CTR นอกจากจะขึ้นกับค่ากระแสอินพุทและเอ้าท์พุทขณะทำงานแล้วยังจะขึ้นกับแรงดันที่ป้อนให้โฟโตทรานซิสเตอร์เอ้าท์พุทด้วย

รูปที่ 2 เป็นกราฟแสดงค่ากระแสเอ้าท์พุท (Icb) ของโฟโตทรานซิสเตอร์ ต่อกระแสอินพุท (If) ของ IRED ของตัวเชื่อมต่อผ่านแสงโฟโตทรานซิสเตอร์เอ้าท์พุทแบบทั่วไป ขณะที่แรงดันตกคร่อมขาคอลเลคเตอร์และขาเบส (Vcb 10 V)

คุณสมบัติอื่นที่สำคัญที่มีการระบุไว้ ได้แก่

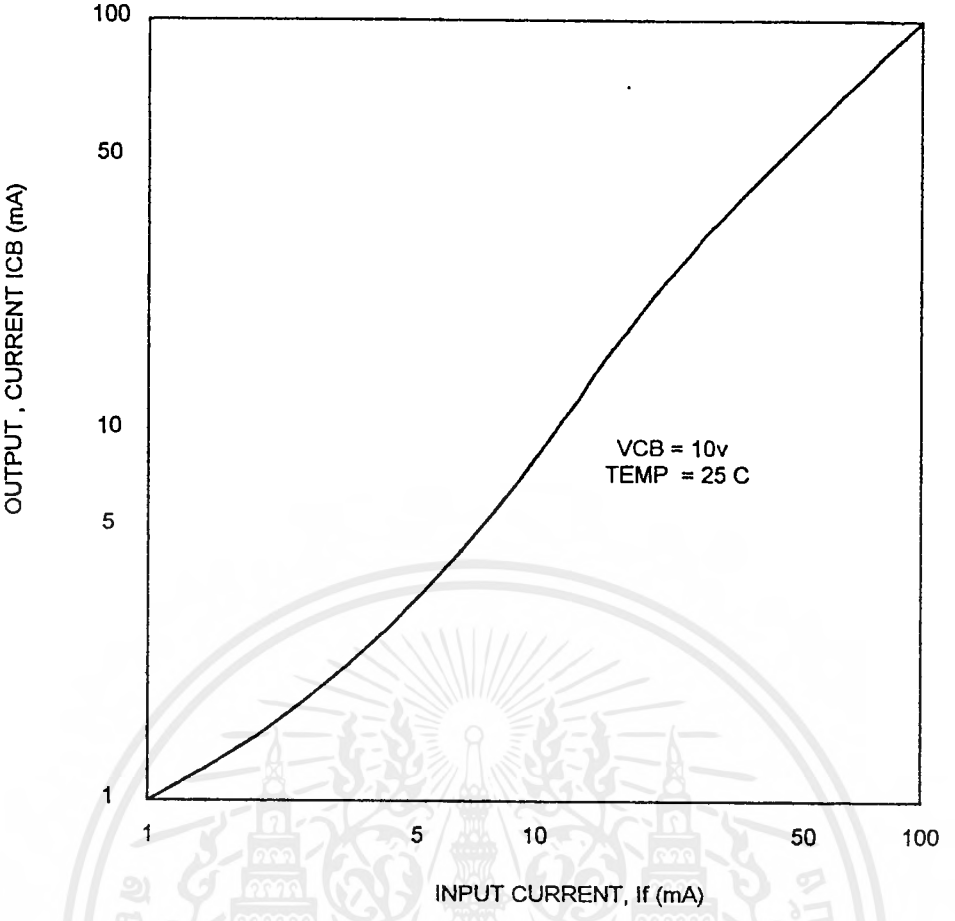
* Isotage Voltage (Viso) (ค่าแรงดันแบ่งแยก) เป็นค่าแรงดันไฟสลับ (AC) สูงสุดที่สามารถตกคร่อมระหว่างอินพุทและเอ้าท์พุทของวงจรโดยยังไม่ทำอันตรายต่ออุปกรณ์ สำหรับตัวเชื่อมต่อผ่านแสงเอ้าท์พุทโฟโตทรานซิสเตอร์ จะมีค่า 500 V -50 kV RMS

* Vce เป็นค่าแรงดันกระแสตรงสูงสุดที่สามารถตกคร่อมที่โฟโตรานซิสเตอร์ส่วนเอ้าท์พุท สำหรับตัวเชื่อมต่อผ่านแสงเอ้าท์พุทโฟโตทรานซิสเตอร์มีค่าอยู่ในช่วง 30-70V

* If ค่ากระแสที่สูงสุดแบบต่อเนื่องที่สามารถไหลผ่าน IRED ได้ สำหรับตัวเชื่อมต่อผ่านแสงเอ้าท์พุทโฟโตทรานซิสเตอร์มีค่า 40-100mA

* Rise/ fall time เป็นค่าเวลาขอบขาขึ้นและขอบขาลงของพัลซ์ซึ่งจะเป็นตัวกำหนดค่าแบบด์วิดซ์ของตัวเชื่อมต่อผ่านแสง สำหรับตัวเชื่อมต่อผ่านแสงเอ้าท์พุทโฟโตทรานซิสเตอร์ จะมีค่า rise และ fall time 2-5 us

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



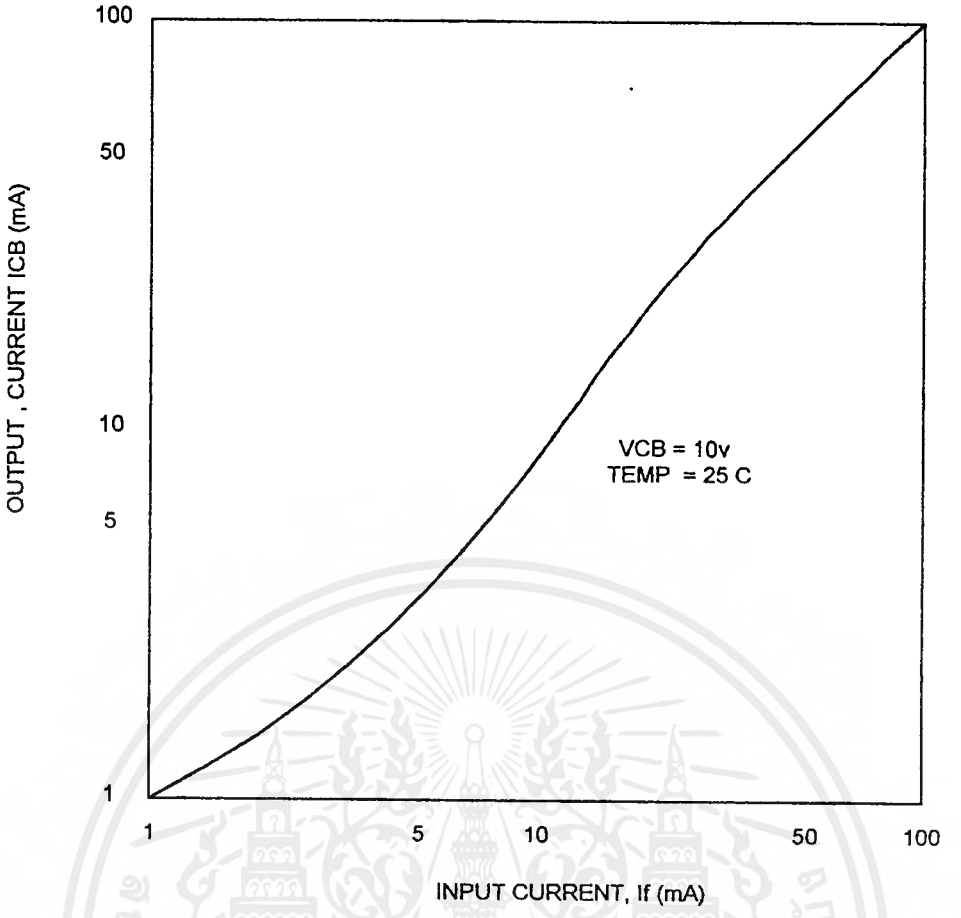
รูปที่ 2 กราฟแสดงค่ากระแสเอ้าท์พุทต่อกระแสดินพุทของตัวชื่อมต่อผ่านแสงเอาท์พุทโฟโตทรานซิสเตอร์แบบทั่วไปที่แรงดัน Vcb 10 V

มาตราฐานทางอุตสาหกรรม

เนื่องจากมีโรงงานที่ผลิตตัวเชื่อมต่อผ่านแสงเป็นจำนวนมาก จึงจำเป็นจะต้องมีการกำหนดให้มีมาตราฐาน เช่น กำหนดลักษณะตัวถังและขาต่อที่เหมือนกัน ดังตัวอย่างที่แสดงในรูปที่ 1 และ 2 ที่เป็นมาตราฐานของตัวเชื่อมต่อผ่านแสงเอ้าท์พุทโฟโตทรานซิสเตอร์แบบ DIP 6 ขา นอกจากนี้ ยังมีบางส่วนที่เป็นแบบ DIP 4 ขา หรือเป็นแบบติดตั้งพื้นผิว

ตัวเชื่อมต่อผ่านแสงแบบหลายแชนเนลที่มีตัวส่งและรับแสงหลายชุดรวมอยู่ในตัวถึงเดียวกัน ที่มีจำหน่ายก็จะเป็นแบบสองและสี่แชนเนล โดยจะมีวงจรเหมือนดังรูที่ 2 หลาย ๆ ชุดมาต่อรวมในตัวถังเดียวกัน ยกเว้นที่ไม่ได้มีการต่อขาเบสออกมาให้ใช้ ตัวเชื่อมต่อผ่านแสงแบบหลายแชนเนลจะมีข้อดีตรงที่ค่าคุณสมบัติทางไฟฟ้าและความร้อนที่ใกล้เคียงกันมาก เนื่องมาจากการที่ต้องวางตัวชิบของตัวส่ง และรับลงในตัวถังเดียวกันที่อยู่ใกล้กันมาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2 กราฟแสดงค่ากระแสเอ้าท์พุทต่อกระแสดินพุทของตัวชื่อมต่อผ่านแสงเอาท์พุทโฟโตทรานซิสเตอร์แบบทั่วไปที่แรงดัน Vcb 10 V

มาตราฐานทางอุตสาหกรรม

เนื่องจากมีโรงงานที่ผลิตตัวเชื่อมต่อผ่านแสงเป็นจำนวนมาก จึงจำเป็นจะต้องมีการกำหนดให้มีมาตราฐาน เช่น กำหนดลักษณะตัวถังและขาต่อที่เหมือนกัน ดังตัวอย่างที่แสดงในรูปที่ 1 และ 2 ที่เป็นมาตราฐานของตัวเชื่อมต่อผ่านแสงเอ้าท์พุทโฟโตทรานซิสเตอร์แบบ DIP 6 ขา นอกจากนี้ ยังมีบางส่วนที่เป็นแบบ DIP 4 ขา หรือเป็นแบบติดตั้งพื้นผิว

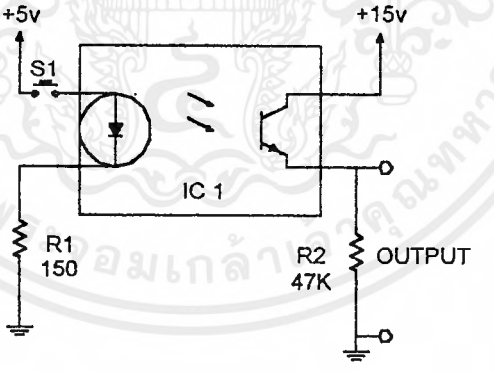
ตัวเชื่อมต่อผ่านแสงแบบหลายแชนเนลที่มีตัวส่งและรับแสงหลายชุดรวมอยู่ในตัวถึงเดียวกัน ที่มีจำหน่ายก็จะเป็นแบบสองและสี่แชนเนล โดยจะมีวงจรเหมือนดังรูที่ 2 หลาย ๆ ชุด มาต่อรวมในตัวถังเดียวกัน ยกเว้นที่ไม่ได้มีการต่อขาเบสออกมาให้ใช้ ตัวเชื่อมต่อผ่านแสงแบบหลายแชนเนลจะมีข้อดีตรงที่ค่าคุณสมบัติทางไฟฟ้าและความร้อนที่ใกล้เคียงกันมาก เนื่องมาจากการที่ต้องวางตัวชิบของตัวส่ง และรับลงในตัวถังเดียวกันที่อยู่ใกล้กันมาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทาง JEDEC ได้กำหนดมาตราฐานของตัวเชื่อมต่อผ่านแสงโฟโตทรานซิสเตอร์เอ้าท์พุทแบบหนึ่งแชนแนลราคาถูก โดยกำหนดให้มีตัวนำเป็น "4N" ที่ได้แก่ 4N25-4N28 และ 4 N35-4N37 และอาจจะมีบางบริษัทได้พัฒนาปรับปรุงบางส่วนเพิ่มและได้ผลิตจำหน่ายในชื่อที่ต่างกันออกไป

เนื่องจากในบางกรณีที่ใช้ตัวเชื่อมต่อผ่านแสงกับวงจรที่ใช้กับไฟบ้าน (AC) จึงต้องมีการทดสอบเรื่องความปลอดภัย โดยมีหน่วยงานทดสอบที่มีชื่อที่ได้แก่ Under-writter Laboratories (UL) และ Canadian Standards Association (CSA) ที่ผู้ผลิตตัวเชื่อมต่อผ่านแสงจะผลิตตามมาตราฐานที่กำหนดให้ นอกจากนี้ อาจมีบางส่วนที่ผลิตตามมาตราฐานของ Verband Beutsh Electrotechniker (VDE) ของประเทศเยอรมัน ซึ่งจำเป็นมากสำหรับสินค้าที่จะจำหน่ายในประเทศแถบยุโรป

รูปที่ 3 เป็นตัวอย่างวงจรของตัวเชื่อมต่อผ่านแสงแบบง่าย ๆ ที่ สามารถควบคุมกระแสของโฟโตทรานซิสเตอร์ได้โดยการควบคุมกระแสที่ไหลผ่าน IRED โดยที่วงจรทั้งสองส่วนนี้แยกจากกัน การทำงานเมื่อ S1 เปิดวงจรไม่มีกระแสไฟลผ่าน IRED IRED ไม่ทำงานไม่มีพลังงานแสงส่งไปที่โฟโตทรานซิสเตอร์ โฟโตทรานซิสเตอร์ไม่ทำงานไม่มีกระแสไหล เป็นเสมือนการเปิดวงจร ทำให้แรงดันตกคร่อมตัวต้านทางเอ้าท์พุท R2 มีค่าเป็น0 โวลท์ เอ้าท์พุทเป็น 0 โวลท์ เมื่อ S1 ปิดวงจรมีกระแสไหลผ่าน IRED และ R1 ทำให้มีแสงอินฟาเรดตกกระทบโฟโตทรานซิสเตอร์ ทำให้นำกระแส มีค่าแรงดันเอ้าท์พุทตกคร่อม R2

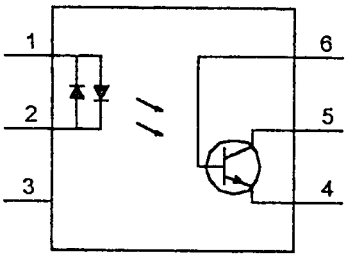


รูปที่ 3 แสดงวงจรการต่อใช้งานตัวเชื่อมต่อผ่านแสงเอ้าท์พุทโฟโตทรานซิสเตอร์

จากวงจรอย่างง่ายในรูปที่ 4 ที่ทำงานแค่เปิดปิดสัญญาณเท่านั้น (หรือทำงานเป็นวงจรดิจิตอลที่มี 2 สถานะ) แต่จะสามารถปรับปรุงวงจรให้ทำงานในแบบอนาลอกที่รับอินพุทและให้

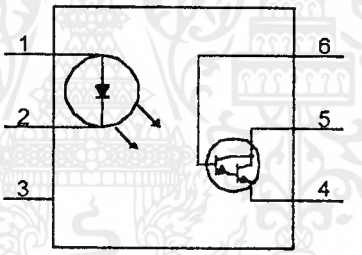
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอ้าท์พุทเป็นสัญญาณต่อเนื่องได้ ซึ่งในกรณีนี้ตัวโฟโตทรานซิสเตอร์ก็จะมีอัตราขยาย ดังที่จะอธิบายในส่วนท้ายของบทความนี้



รูปที่ 4 ผังวงจรของตัวเชื่อมต่อผ่านแสงเอ้าท์พุทโฟโตทรานซิสเตอร์ที่อินพุทเป็นกระแสสลับ (AC INPUT)

ตัวเชื่อมต่อผ่านแสงที่อินพุทเป็น IRED ส่วนเอ้าท์พุทเป็นตัวตรวจจับแสงแบบต่าง ๆ รูปที่ 5 เป็นตัวเชื่อมต่อผ่านแสงเอ้าท์พุทโฟโตทรานซิสเตอร์ที่รับอินพุทได้สองทิศทาง ที่อินพุทเป็น IRED แบบเกลเลี่ยมอาร์เซไนด์สองตัวที่ขนานสลับข้าง เพื่อให้สามารถเชื่อมโยงสัญญาณกระแสสลับได้หรือป้องกันอินพุทจากการต่อกลับขั้ว โดยจะมี CTR ค่าต่ำสุดโดยทั่วไป 20%



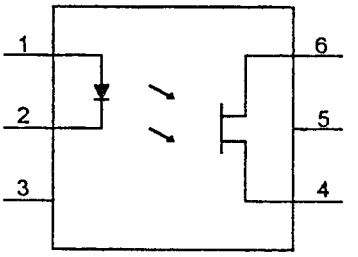
รูปที่ 6 ผังวงจรของตัวเชื่อมต่อผ่านแสงเอ้าท์พุทโฟโตดาร์ลิงตันทรานซิสเตอร์

รูปที่ 6 เป็นตัวเชื่อมต่อผ่านแสงเอ้าพุทโฟโตดาร์ลิงตันทรานซิสเตอร์ ที่มีค่าอัตราขยายที่สูงมาก ทำให้ได้กระแสเอ้าท์พุทมากกว่าแบบโฟโตทรานซิสเตอร์ธรรมดา โดยทั่วไปจะมี CTR ค่าต่ำสุด (ที่แรงดันตกคร่อมขาคอลแลคเตอร์กับอิมิตเตอร์ที่ 30-35 V) ถึง 500% ซึ่งมากเป็น 10 เท่าของแบบโฟโตทรานซิสเตอร์ธรรมดา แต่ก็มีข้อเสียตรงที่ความไวในการทำงานก็จะลดลง 10 เท่าตามไปด้วย ทำให้ไม่สามารถใช้กับวงจรที่มีความเร็วสูงได้

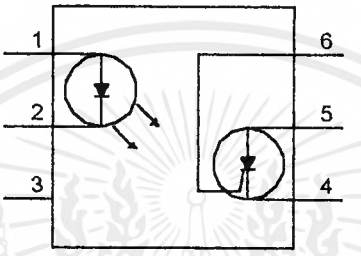
ตัวเชื่อมต่อผ่านแสงที่เอ้าท์พุทเป็นไทริสเตอร์จะมีอยู่สองแบบดังที่แสดงในรูปที่ 7 และ 8 โดยในรูปที่ 7 เป็นแบบเอ้าท์พุทเป็นโฟโตเอสซีอาร์ (photo SCR หรือ OptoSCR) โดยทั่วไปจะมีค่า isolating voltage 1000-4000 VRMS, blocking voltage ต่ำสุด 200-400V ค่ากระแสเทริน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

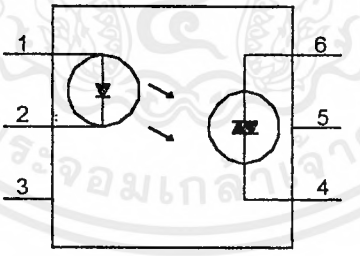
ออนสูงสุด (maximum turnon current (Ift) 10 mA ส่วนรูปที่8 เป็นแบบเอ้าท์พุทโฟโตไตรแอค ส่วนของไทริสเตอร์เอ้าท์พุทโดยทั่วไปจะมีค่า forward blocking voltage (Vdrm) ที่ 400V



รูปที่ 7 ผังวงจรของตัวเชื่อมต่อผ่านแสงเอ้าท์พุทโฟโตเอสซีอาร์



รูปที่ 8 ผังวงจรของตัวเชื่อมโยงทางแสงเอ้าท์พุทโฟโตเอสซีอาร์



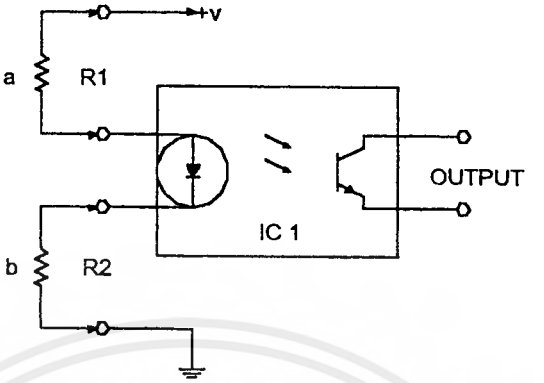
รูปที่ 9 ผังวงจรของตัวเชื่อมต่อผ่านแสงเอ้าท์พุทโฟโตไตรแอค

ตัวอย่างการประยุกต์ใช้งานตัวเชื่อมต่อผ่านแสง

การต่อตัวเชื่อมต่อผ่านแสงในวงจรก็จะเหมือนกับการต่อวงจรที่มีตัว LED กำเนิดแสง และวงจรของโฟโตทรานซิสเตอร์รับแสงที่ต่อแยกจากกัน ในการใช้งานจะต้องต่อตัวต้านทาน

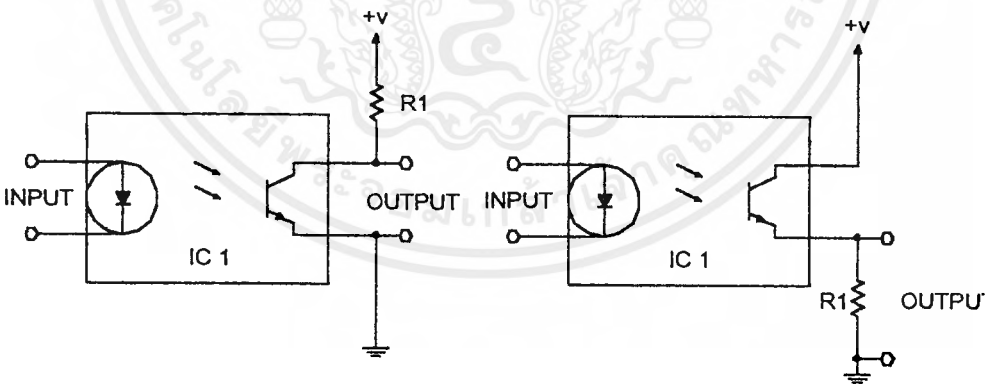
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เพื่อจำกัดกระแสของ IRED ไม่ให้มากเกิน ซึ่งจะต้องต่ออนุกรมกับวงจรของ IRED ได้สองแบบดังที่แสดงในรูปที่ 10 รูป (ก) เป็นการต่อที่ขาอาโนดของ IRED ส่วนรูป (ข) ต่อที่ขาคาโถด โดยต้องเลือกต่อแบบใดแบบหนึ่ง



รูปที่ 10 แสดงการต่อตัวต้านทานเพื่อจำกัดกระแสของ IRED ที่รูป ก เป็น R1 หรือต่อตามรูป ข เป็นรูป R2

จากการทำงานของตัวเชื่อมต่อผ่านแสงที่ให้เอาท์พุทเป็นกระแสนี้สามารถเปลี่ยนเอ้าท์พุทให้เป็นแรงดันได้ ด้วยการต่อตัวต้านทานภายนอกต่ออนุกรมกับขาคอลเลคเตอร์หรือขาอีมิตเตอร์ดังที่แสดงในรูปที่ 11 (ก) ต่อที่ขาคอลเลคเตอร์ รูป 11 (ข) ต่อที่ขาอีมิตเตอร์ ค่าความไวของวงจรจะแปรผันตรงกับค่าของตัวต้านทานที่ต่ออนุกรมนี้

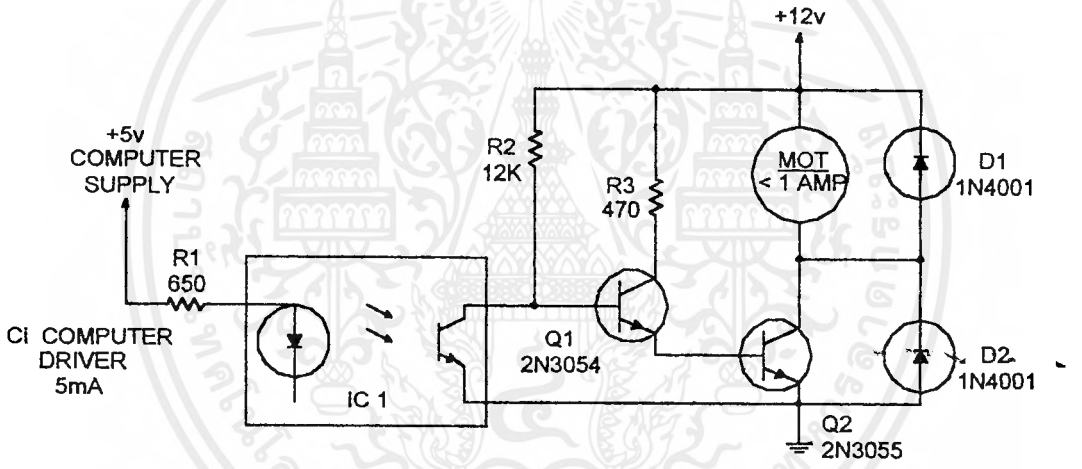


รูปที่11 แสดงการต่อตัวต้านทานภายนอกโดยรูป ก. ต่อที่ขาคอลเลคเตอร์รูป ข.ต่อที่ขาอีมิตเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการใช้งานสามารถเปลี่ยนเอาท์พุทจากโฟโตทรานซิสเตอร์ให้เป็นโฟโตไดโอดได้ โดยจะทำได้เฉพาะกับแบบที่อยู่ในตัวถังแบบ DIP 6 ขาเท่านั้น โดยดัดแปลงได้โดยย้ายเอาท์พุทที่เคยต่อที่ขาคอลเลคเตอร์หรืออีมิตเตอร์มาต่อที่ขาเบส (ขา 6) แทน ส่วนขาอีมิตเตอร์ ให้ปล่อยว่างไว้หรือจะต่อเข้ากับขาเบสก็ได้ การต่อในแบบนี้จะทำให้ลดค่าเวลาขาขึ้นของสัญญาณอินพุทได้มากทำให้สามารถใช้งานกับสัญญาณอินพุทความเร็วมาก ๆ ได้ แต่ก็จะเป็นการลดค่า CTR ลงอย่างมากโดยจะลดลงเหลือประมาณ 0.2 %

รูปที่ 12 แสดงการใช้ตัวเชื่อมต่อผ่านแสงเอาท์พุทโฟโตทรานซิสเตอร์ต่อเชื่อมโยงระหว่างเอ้าท์พุทที่เป็นดิจิตอลของเครื่องคอมพิวเตอร์ (5V, 5mA) กับมอเตอร์กระแสตรงแรงดัน 12 V ที่ขณะทำงานจะกินกระแสน้อยกว่า 1 A การทำงานขณะที่เอาท์พุทของคอมพิวเตอร์มีลอจิก 1 IRED และ โฟโตทรานซิสเตอร์จะไม่ทำงาน ทำให้ Q1 และ Q2 จะทำงานทำให้มีกระแสไหลผ่านมอเตอร์มอเตอร์ทำงาน เมื่อเอาท์พุทเป็นลอจิก 0 IRED และโฟโตทรานซิสเตอร์จะทำงานทำให้ Q1 และ Q2 และมอเตอร์ไม่ทำงาน วงจรนี้มีข้อจำกัดที่มอเตอร์จะต้องกินกระแสไม่เกิน 1 A



รูปที่ 12 แสดงการนำเอาตัวเชื่อมต่อผ่านทางแสงเอ้าท์พุทโฟโตทรานซิสเตอร์ ไปต่อเชื่อมโยงจากเอ้าท์พุทของคอมพิวเตอร์กับมอเตอร์กระแสตรง

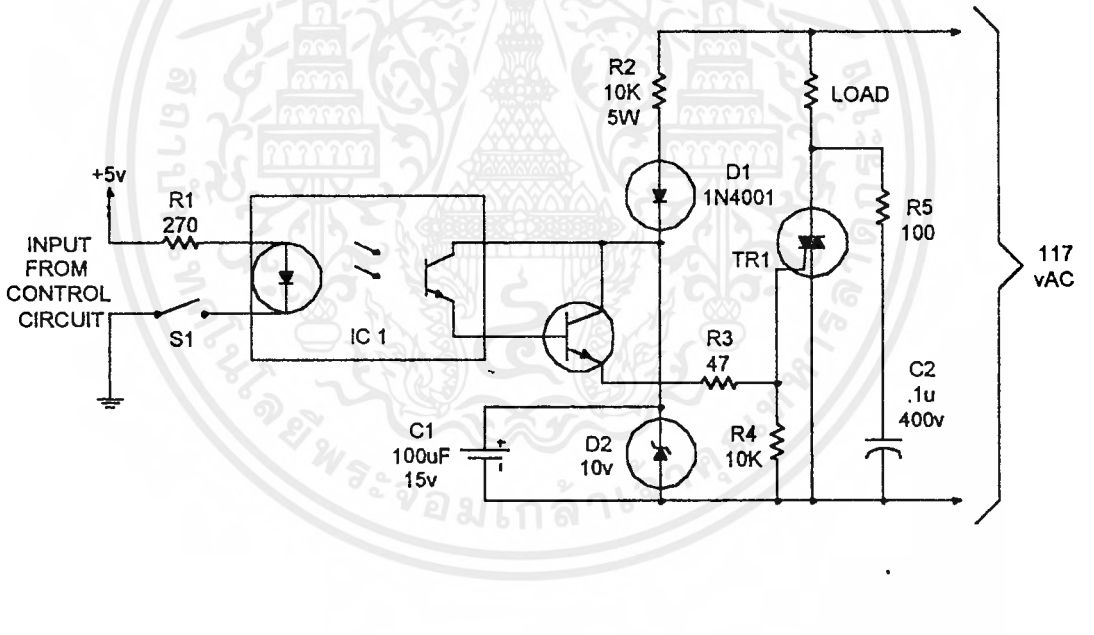
การเชื่อมโยงกับไตรแอค

ตัวเชื่อมต่อผ่านแสงจะเหมาะสมกับการใช้ต่อเชื่อมโยงเอาท์พุทจากวงจรควบคุมที่มีค่าแรงดันต่ำไปยังวงจรควบคุมไฟกระแสสลับแรงดันทางที่ใช้ไตรแอคควบคุม เนื่องจากข้อดีของ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

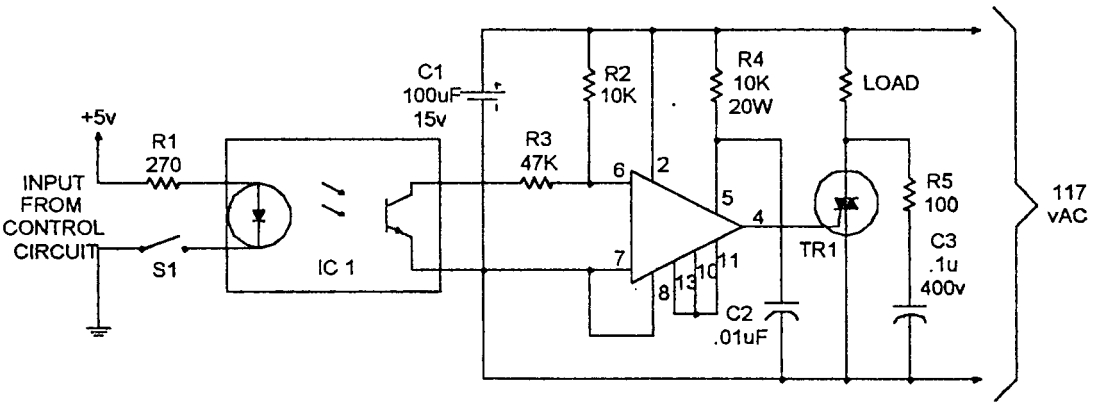
การที่แบ่งแยกวงจรทั้งสองส่วนออกจากกันอย่างเด็ดขาด โดยจะมีการจัดวงจรได้ดังรูปที่ 13 ที่สามารถควบคุมหลอดไฟ ฮีตเตอร์ มอเตอร์ หรือโหลดประเภทอื่นได้

รูปที่ 14 และ 15 เป็นวงจรควบคุมที่ใช้งานจริง ตัวไตรแอคที่ใช้จะต้องเลือกให้ทนแรงดันและกระแสให้เหมาะสมกับโหลดที่ใช้ วงจรในรูปที่ 19 จะเป็นแบบ non-synchronous switching คือการสวิทซ์ของตัวไตรแอคจะไม่สัมพันธ์กับรูปคลื่นแรงดัน AC อินพุท ซึ่งถ้าไตรแอคไปสวิทซ์ขณะที่แรงดัน AC มีค่าสูงสุด จะทำให้แรงดันมีการเปลี่ยนแปลงทันทีทันใด เกิดกระแสกระชาก ซึ่งจะทำให้เกิดสัญญาณรบกวน RFI (radio frequency interference) ไปรบกวนอุปกรณ์อื่น ๆ จากวงจรแรงดัน AC อินพุทที่ผ่าน R2 จะถูกเรดติไฟด้วย D1 ซีเนอร์ไดโอด D2 และ C1 ให้เป็นแรงดันกระแสตรงค่า 10 V แรงดัน 10 V นี้จะเป็นแรงดันที่ป้อนให้ขาเกตของไตรแอค โดยผ่าน Q1 เพื่อควบคุมให้ไตรแอคและโหลดทำงานหรือไม่ทำงาน เมื่อ S1 เปิดวงจร ตัวเชื่อมโยงผ่านแสงไม่ทำงานไม่มีกระแสเบสไหลผ่าน Q1 ทำให้ไตรแอคและโหลดไม่ทำงาน ถ้า S1 ปิดวงจร ตัวเชื่อมต่อผ่านแสงจะทำงานมีกระแสเบสไหลผ่าน Q1 ทำให้ Q1 ทำงานต่อแรงดัน 10 V ผ่าน R3 เข้ากับขาเกตของไตรแอค ทำให้ไตรแอคทำงานต่อโหลดเข้ากับไฟ AC



รูปที่ 13 วงจรควบคุมไฟเอซีโดยใช้ TRIAC แบบ NON-SYNCHRONOUS ที่ส่วนอินพุทเป็นตัวเชื่อมต่อผ่านแสง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

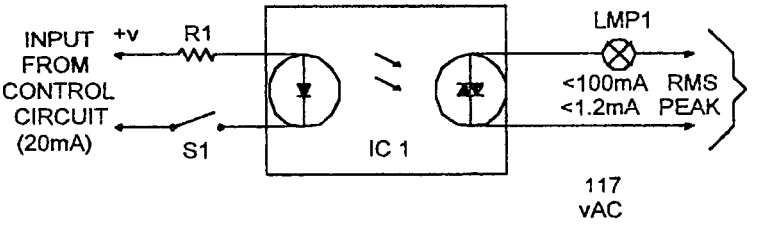


รูปที่ 14 วงจรควบคุมไฟเอซีโดยใช้ TRIAC แบบ SYNCHRONOUS ที่ส่วนอินพุทเป็นตัวเชื่อมต่อผ่านแสง

วงจรในรูปที่ 14 จะเป็นแบบที่ synchronous power switching ที่มีอุปกรณ์ที่สำคัญคือ ไอซี CA3059/3079 silicon monolithic zero-voltage switch ของบริษัท Motorola และ Harris ผลจากการต่อไอซีนี้เข้ากับเอาท์พุทของตัวเชื่อมต่อผ่านแสงที่เป็นโฟโตทรานซิสเตอร์ จะทำให้วงจรเป็นแบบ synchronous power switching ที่จะมีกระแสเกตไปกระตุ้นให้ไตรแอคทำงานเมื่อค่แรงดัน AC ขณะนั้นมีค่าศูนย์หรือเกือบศูนย์โวลท์ (หรือกระตุ้นตรงจุดที่รูปคลื่นของแรงดันตัดผ่านศูนย์ : zero crossing) วงจรในลักษณะนี้จะมีใช้กันมากในตัว solid-state relay สำเร็จรูปที่มีจำหน่ายกันทั่วไป

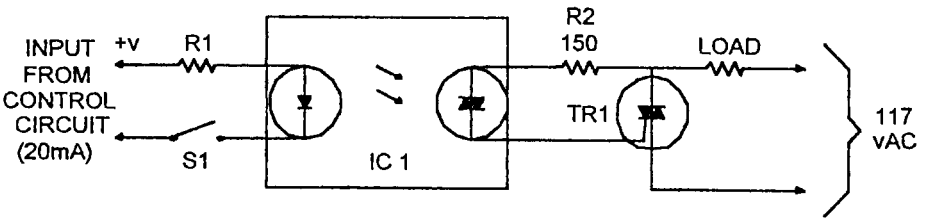
ตัวเชื่อมต่อผ่านแสงเอาท์พุทโฟโตเอสซีอาร์และโฟโตไตรแอค

ตัวโฟโตเอสซีอาร์และโฟโตไตรแอคที่ส่วนเอาท์พุทของตัวเชื่อมต่อผ่านแสงจะมีค่าอัตราทนกระแสเอาท์พุทที่ไม่มากนัก แต่ก็จะมีอัตราทนกระแสกระชาก (surge-current) ได้มากกว่าค่า RMS ที่ทนได้อย่างในกรณีของเอสซีอาร์ที่พัลซ์กว้าง 100 us มีค่าดิวตี้ไซเกิ้ล (duty-cycle) น้อยกว่า 1 % ก็จะทนกระแสกระชากได้ถึง 5 A ส่วนของไตรแอคก็จะทนกระแสกระชากได้ถึง 1.2 A ที่พัลซ์กว้าง 10 us มีดิวตี้ไซเกิ้ลสูงสุด 10 %



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 15 วงจรควบคุมหลอดไฟแบบมีไส้โดยใช้ตัวเชื่อมต่อผ่านแสงเอาท์พุทโฟโตไตรแอค



รูปที่ 16 วงจรที่ได้ดัดแปลงเพิ่มเติมจากวงจรในรูปที่ 21 ให้สามารถใช้งานกับโหลดที่มีกำลังสูง ๆ ได้

ในการทำงานของตัวเชื่อมต่อผ่านแสงเอาท์พุทโฟโตเอสซีอาร์และโฟโตไตรแอคที่ส่วน IRED อินพุทก็จะมีการใช้และการจัดวงจรเหมือนกับตัวเชื่อมต่อผ่านแสงเอาท์พุทโฟโตทรานซิสเตอร์ทั่วไป ส่วนการต่อโฟโตเอสซีอาร์และโฟโตไตรแอคก็จะต่อเหมือนกับการต่อเอสซีอาร์และไตรแอคทั่วไป แต่ต้องระวังเรื่องอัตราการทนกระแสของตัวอุปกรณ์เอง รูปที่ 15, 16 และ 17 แสดงวงจรใช้งานจริงของตัวเชื่อมต่อผ่านแสงเอาท์พุทโฟโตไตรแอค ค่า R1 ในวงจรทุกรูปจะต้องเลือกให้มีค่าที่ทำให้กระแสไหลผ่าน IRED มีค่าอย่างน้อย 200mA

ในรูปที่ 15 จะต่อโฟโตไตรแอคตรงกับไฟ AC เพื่อควบคุมหลอดไฟแบบไส้ไม่เกิน 100mARMS และมีค่ากระแสพุ่งเข้า (inruch current) สูงสุดน้อยกว่า 1.2A

รูปที่ 22 รูปที่ 16เป็นวงจรที่ปรับปรุงขึ้น ด้วยการนำเอ้าท์พุทของโฟโตไตรแอคของตัวเชื่อมต่อผ่านแสงไป ทริกไตรแอคตัวลูกอีกตัว ที่สามารถเลือกให้ทนกระแสและแรงดันค่าต่าง ๆ ตามที่ต้องการ วงจรในรูปนี้ใช้งานเฉพาะกับโหลดแบบไม่มีค่าความเหนี่ยวนำ (เป็นโหลดที่มีแต่ค่าความต้านทานเท่านั้น) เช่น โหลดที่เป็นหลอดไฟแบบไส้หรืออีตเตอร์

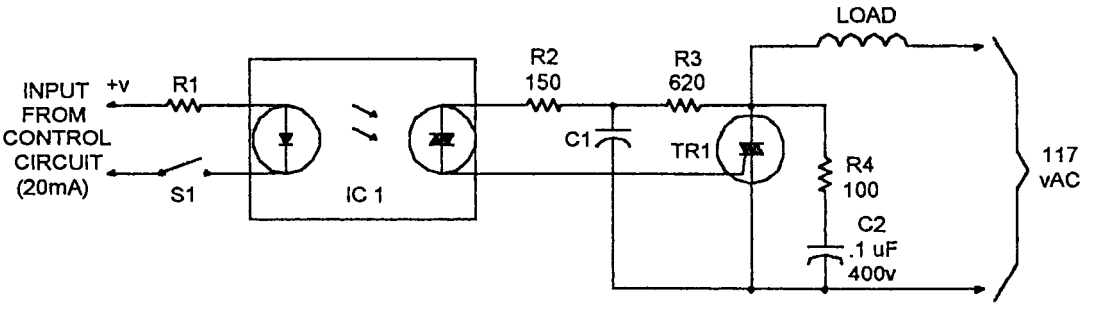
รูปที่ 17 เป็นการนำวงจรในรูปที่ 22 มาดัดแปลงให้สามารถใช้งานกับโหลดที่มีค่าความเหนี่ยวนำ เช่น มอเตอร์ ด้วยการเพิ่ม R2, C1 และ R3 ที่เป็นวงจรเลื่อนเฟสของกระแสที่ไปทริกขาเกตของไตรแอคให้ทำงานในเวลาที่ถูกต้อง ตัวต้านทาน R4 และ C2 ต่อเป็นวงจรสนับเบอร์เพื่อลดผลของกระแสกระชาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางคุณสมบัติของเบอร์ 4N26

		ค่าต่ำสุด	ค่าปกติ	ค่าสูงสุด	หน่วย
อินพุท	IF			80	mA
	VF(IF=10mA)		1.1	1.5	V
	VR			3	V
เอาท์พุท	IC			100	mA
	V(BR)CBO	70			mV
	V(BR)CEO	30			mV
	V(BR)ECO	7			mV
ตัวแปรเชื่อมโยง	CTR(IF=10mA,VCF=10 V)	20%			
	VCE(SET)(IF=50mA,IC=2mA)		0.1	0.5	V

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 17 วงจรควบคุมโหลดที่มีค่าความเหนี่ยวนำ โดยใช้ตัวเชื่อมต่อทางแสงเอ้าทพุทโฟโตไตรแอคและไตรแอคตัวลูก (TR1)

การทดสอบจะทำการสั่งงานทางคอมพิวเตอร์โดยส่งข้อมูลผ่านทาง**RS 232**หรือ **RS 485** การกำหนดโปรโตคอล

ในระบบ SMALL SCAL DCS มีดังนี้

General Instrument Protocol (GITP)

การกำหนดโปรโตคอล (Protocol) ในการติดต่อสื่อสารระหว่าง คอมพิวเตอร์หลัก (Host Computer) กับ Controller Module ต่างๆ เราได้ออกแบบ Protocol ขึ้นมาใช้เองโดยมีข้อกำหนด คือ ต้องง่ายไม่ซับซ้อน ,ขนาดของ Packet ข้อมูลเหมาะสมไม่ยาวจนเกินไป และ มีความยืดหยุ่นสูง สามารถ ประยุกต์ใช้ ได้อย่างกว้างขวาง โดยใช้ชื่อว่า General Instrument Protocol (GITP)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางข้อกำหนดของ GITP

Packet byte Number	Packet type	Character available
0	Start	!
1	ST.No	0-9
2	ST.No	0-9
3	General Command & Status	A-Z
4	DATA	0-F
5	DATA	0-F
6	General purpose	All
7	General purpose	All
8	Check Sum	All
9	End	#

GITP มีขนาดของ DATA Packet เท่ากับ 10 Byte แบ่งออกเป็น 7 ส่วน แต่ละ Byte เก็บข้อมูล อักขระ ASCII 1 ตัว มีข้อกำหนดต่างๆ ดังตาราง แยกอธิบายได้ดังนี้

ส่วนที่ 1 "START" ประกอบด้วย ข้อมูล 1 Byte คือ Byte 0 เป็นส่วนที่ทำหน้าที่บอกสถานะเริ่มต้นของ Packet อักขระที่ใช้ได้มีเพียงตัวเดียว คือ "!"

ส่วนที่ 2 "ST.No." ประกอบด้วย ข้อมูล 2 Byte คือ Byte 1,2 เป็นส่วนที่ทำหน้าที่บอกหมายเลขของอุปกรณ์ที่ติดต่อ อักขระที่ใช้ได้คือตัวเลข "0-9"

ส่วนที่ 3 "General Command & Status" ประกอบด้วย ข้อมูล 1 Byte คือ Byte 3 เป็นส่วนที่กำหนดคำสั่ง และหรือ บอกสถานะ ของอุปกรณ์ อักขระที่ใช้ได้ คือ "A-Z"

ส่วนที่ 4 "DATA" ประกอบด้วย ข้อมูล 2 Byte คือ Byte 4,5 เป็นส่วนที่ทำหน้าที่ส่งผ่านค่า Input,Output ของอุปกรณ์ อักขระที่ใช้ได้คือ "0-F"

ส่วนที่ 5 "General purpose" ประกอบด้วย ข้อมูล 2 Byte คือ Byte 6,7 เป็นส่วนที่สามารถเขียนโปรแกรมนำไปใช้อย่างไรก็ได้ สามารถใช้อักขระได้ทุกตัว

ส่วนที่ 6 "Check Sum" ประกอบด้วย ข้อมูล 1 Byte คือ Byte 8 เป็นส่วนที่ใช้สำหรับตรวจสอบความถูกต้องของ DATA Packet สร้างจากการนำ ข้อมูล Byte 1-7 ใน Packet นั้นๆ มาบวกกันโดยไม่คิดตัวทด สามารถใช้อักขระได้ทุกตัว

ส่วนที่ 7 "END" ประกอบด้วย ข้อมูล 1 Byte คือ Byte 9 เป็นส่วนที่ทำหน้าที่บอกการสิ้นสุดของ Packet อักขระที่ใช้ได้มีเพียงตัวเดียว คือ "#"

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมการควบคุมและการติดต่อกับคอมพิวเตอร์ที่ใช้ขับสเต็ปมอเตอร์4บิต ถูกโปรแกรมลงในอีพรอมมูเลเตอร์

ตัวอย่างการทดสอบโดยการส่ง PACKET ของGITPไปควบคุมการหมุนของสเต็ปมอเตอร์

การส่งPACKETควบคุม

สเต็ปมอเตอร์ให้หมุนทางขวาโดยการป้อนข้อมูลตามPACKET1 ข้อมูลในส่วนที่3จะถูกป้อนด้วยอักขระRในส่วน

ของDATAจะเป็นส่วนกำหนดการหมุนช้าหรือหมุนเร็วโดยสามาร ถป้อนDATAได้ตั้งแต่00-FF

Byte	0	1	2	3	4	5	6	7	8	9
	!	0	6	R	0	0	+	+	s	#
	START	ST.No.		Cmd,Status	DATA		General purpose		ChkSum	END

รูป PACKET ที่ 1

การส่งPACKETไปควบคุมสเต็ปมอเตอร์ให้หมุนซ้ายทำได้โดยการป้อนข้อมูลตามPACKETที่2โดยส่วนที่3ของ PACKET จะป้อนอักขระส่วนDATAก็จะเป็นตัวกำหนดการหมุนเช่นเดียวการหมุนขวา

Byte	0	1	2	3	4	5	6	7	8	9
	!	0	6	L	0	0	+	+	s	#
	START	ST.No.		Cmd,Status	DATA		General purpose		ChkSum	END

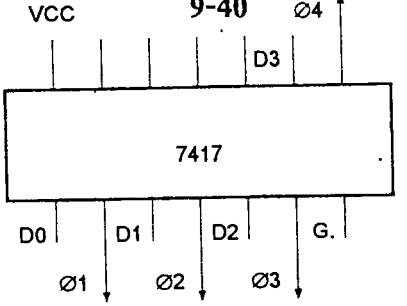
รูป PACKET ที่ 2

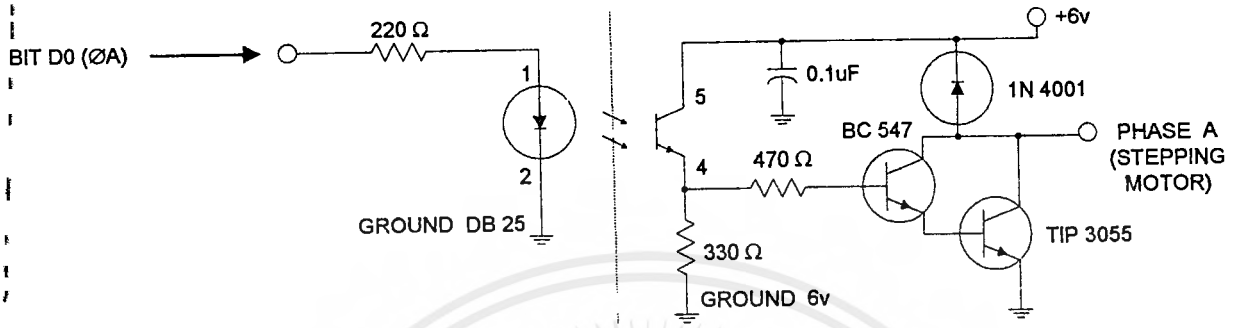
การควบคุมให้สเต็ปมอเตอร์หยุดการทำงานสามารถทำได้โดยการป้อนอักขระใดๆที่มิใช่ R,Lลงใน PACKETส่วน ที่3

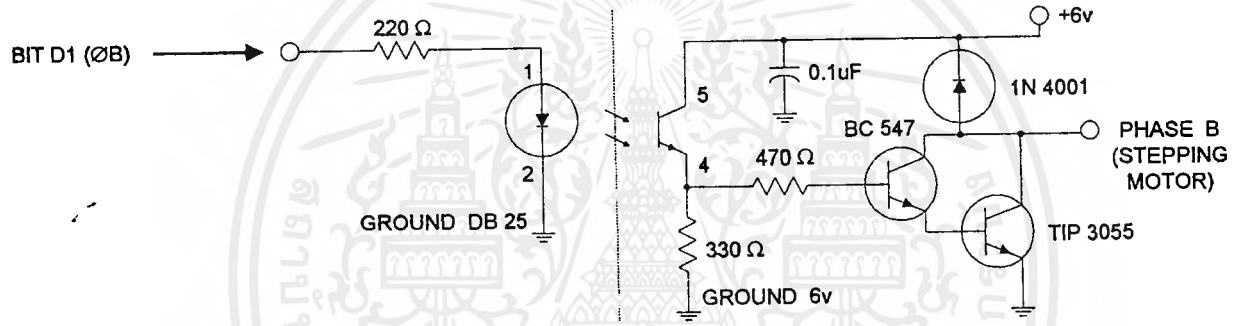
Byte	0	1	2	3	4	5	6	7	8	9
	!	0	6	S	0	0	+	+	s	#
	START	ST.No.		Cmd,Status	DATA		General purpose		ChkSum	END

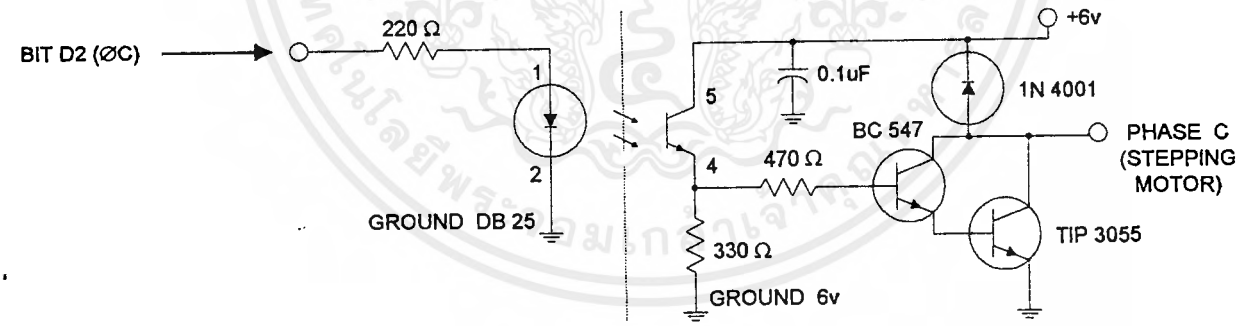
รูป PACKET ที่ 3

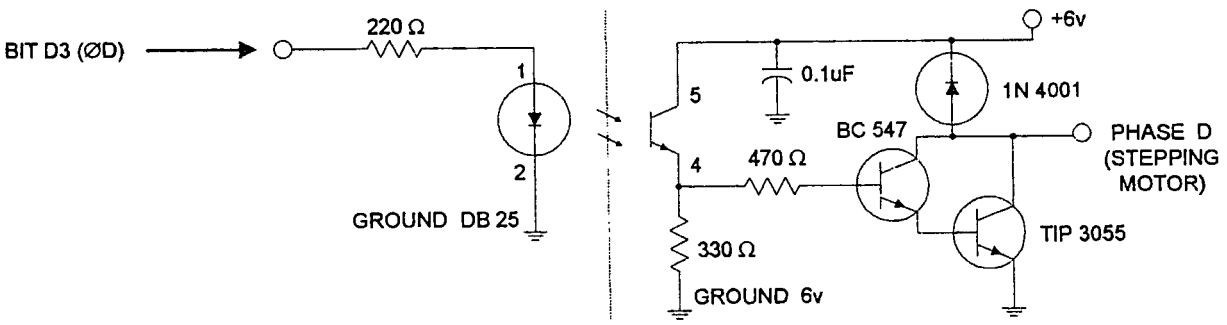
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้





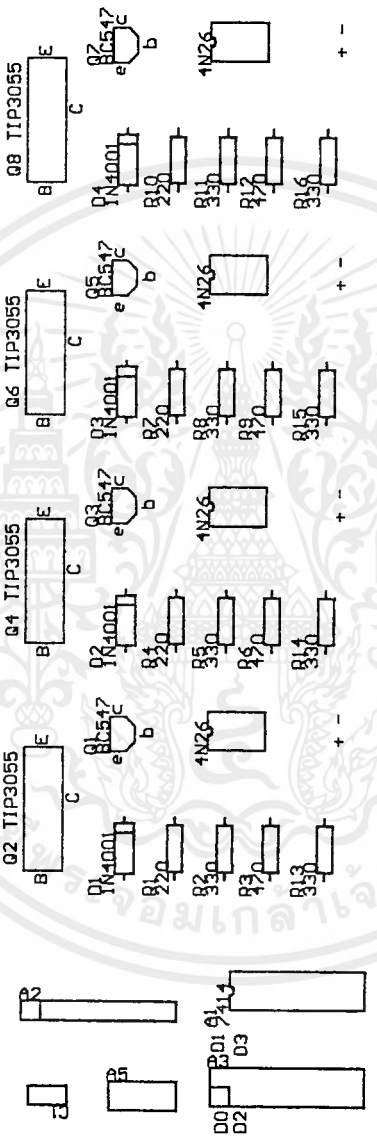






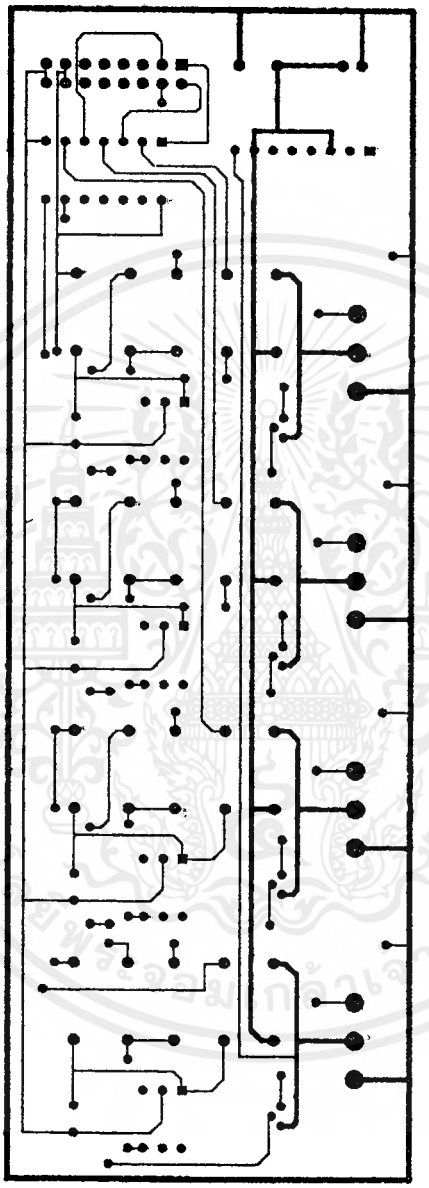
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ลายวงจรชุดขับสเต็ปมอเตอร์



การลงอุปกรณ์ชุดขับสเต็ปมอเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ลายทองแดงชุดขับสเต็ปมอเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

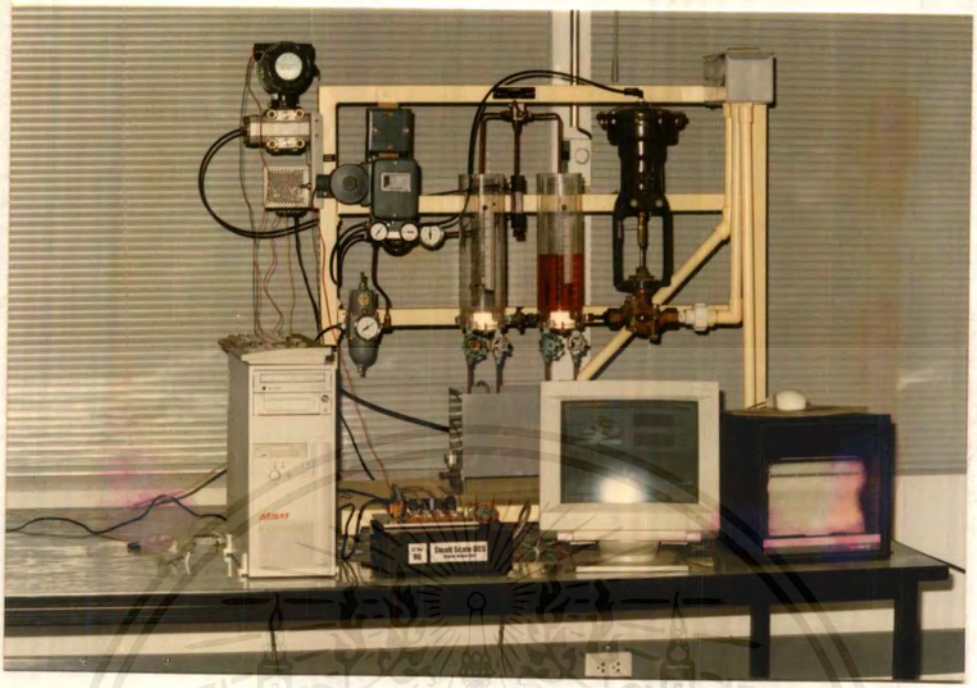
ปัญหาและอุปสรรค

ในการทดลอง digital output unit ปัญหาที่พบแบ่งออกได้เป็น2ส่วนคือ

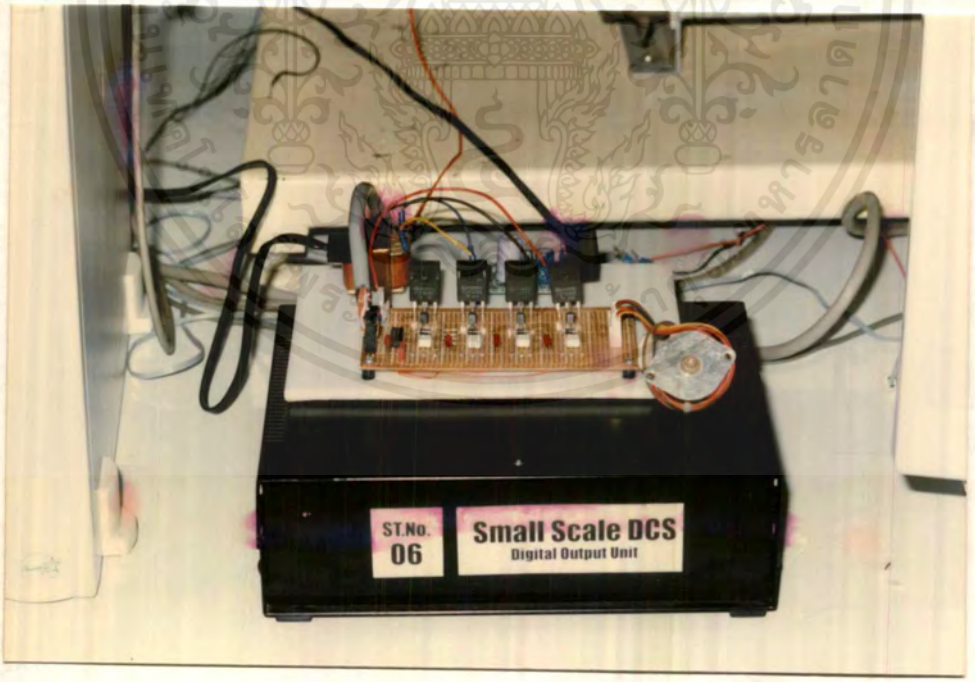
ส่วนที่1 ในส่วนของโปรแกรมที่ใช้ในการอินเตอร์เฟสกับคอมพิวเตอร์เพื่อควบคุมการทำงานของสเต็บมอเตอร์

ส่วนที่2 คือส่วนฮาร์แวร์ของชุดขับสเต็ปมอเตอร์ปัญหาที่พบคือช่วงแรกที่ทำการทดลองปรากฏว่าสเต็ปมอเตอร์ไม่สามารถทำงานได้ทัน การแก้ปัญหาในส่วนนี้คือได้นำ IC TTLเบอร์ 7417 ซึ่งเป็น ICบับเฟอร์มาต่อเข้ากับชุดขับสเต็ปมอเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

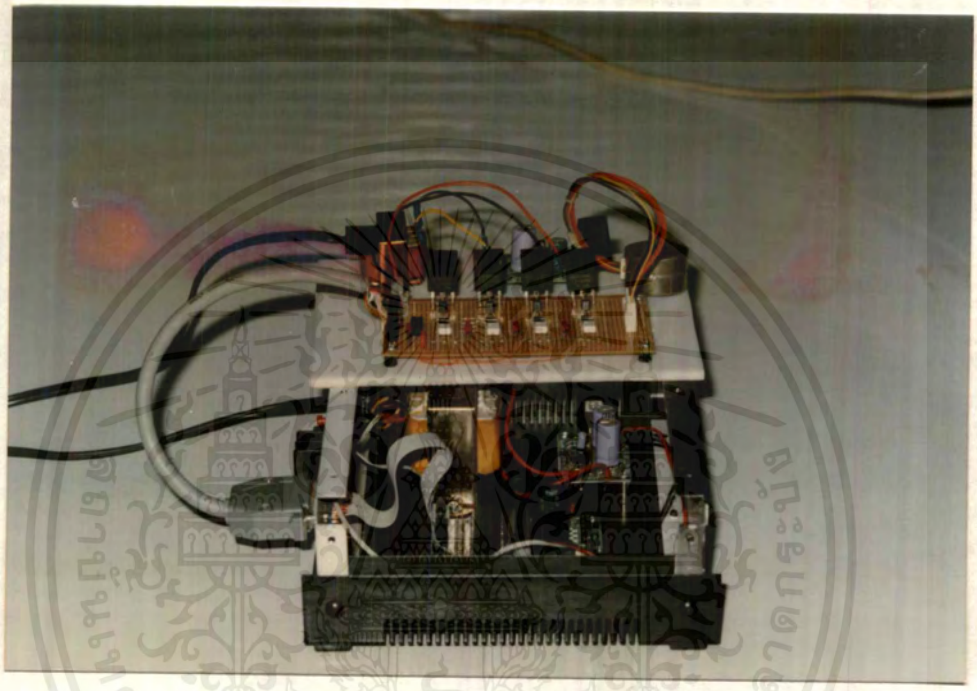


รูปแสดง UNIT 06 (DIGITAL OUTPUT UNIT) กับการควบคุมกระบวนการระดับน้ำ



รูปแสดง UNIT 06 (DIGITAL OUTPUT UNIT) และชุดทดลองสเต็ปมอเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปแสดงการต่อวงจรภายในของ UNIT 06 และชุดทดลองสเต็ปมอเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปแสดงหน้าจอของ ST.No. 06

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่10

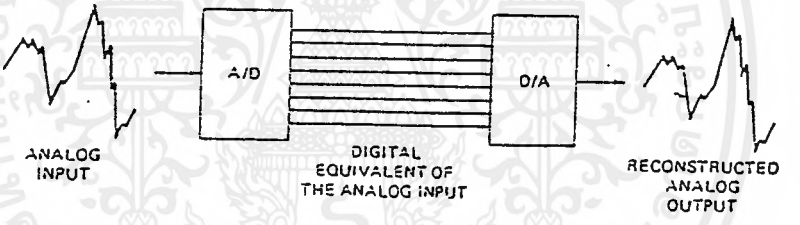
ANALOG I/O , O/P UNIT

10.1 การแปลงสัญญาณดิจิตอลเป็นสัญญาณอนาล็อก

(DIGITAL TO ANALOG CONVERTER)

ความรู้พื้นฐาน

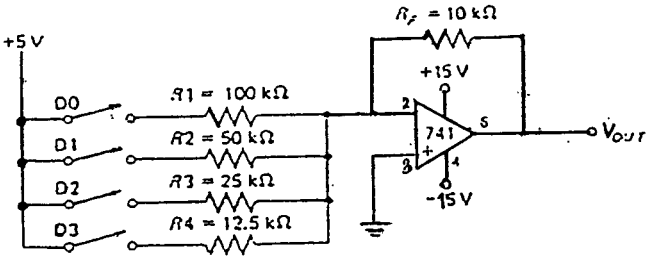
ในโลกทุกวันนี้คอมพิวเตอร์ได้เข้ามามีบทบาทกับสังคมมนุษย์มากขึ้น ดังนั้น จึงมีความต้องการอุปกรณ์ D/A มากขึ้นตามไปด้วย เราจะพบว่า การส่งเสียงดนตรี เสียงพูด และข้อมูล โดยดิจิตอลจะสามารถกำจัดสัญญาณ รบกวนได้มาก และ ส่งสัญญาณในรูปกระแสไฟฟ้า ได้ง่าย อย่างไรก็ตาม ทางเครื่องรับ สัญญาณที่ส่งแบบดิจิตอลจะถูกเปลี่ยนกลับเป็นสัญญาณอนาล็อกดังเดิมอีกครั้ง ดังในรูปที่ 10.1



รูปที่ 10.1 (DIGITAL TO ANALOG CONVERTER)

วงจรเปลี่ยนสัญญาณดีทูเอ ชนิดนี้ใช้ ตัวต้านทานต่าง ๆ และออปแอมป์เพื่อเปลี่ยนระดับสัญญาณลอจิก 2 ระดับเป็นแรงดันที่ได้สัดส่วนกัน รูปที่ 10.2 แสดงวงจรเปลี่ยนสัญาณดิจิตอลขนาด 4 บิต ออปแอมป์ที่ใช้มีอัตราขยายสูงมาก (โดยทั่วไปจะสูงกว่า 100,000 เท่า) มีความต้านทานด้านเอาต์พุตต่ำ ความต้านทานด้านอินพุตมีค่าสูงมาก สิ่งสำคัญที่สุดที่จะต้องตระหนักไว้ ก็คือ สัญญาณที่เอาต์พุตถูกป้อนกลับมายังอินพุตแบบกลับเฟส(การป้อนกลับแบบลบ เพื่อเปรียบเทียบกับสัญญาณที่ขาอินพุต แบบไม่กลับเฟสเอาต์พุตของออปแอมป์จะเป็นตัวจ่าย หรือรับกระแส (Source or sink) เพื่อให้แรงดันที่เปรียบเทียบกันนั้นมีค่าเดียวกัน วงจรในรูปที่ 10.2 ต่อขาไม่กลับเฟสลงกราวด์ ดังนั้นที่ขากลับเฟสก็จะมีแรงดัน 0 โวลต์ด้วย การที่อินพุตที่ขากลับเฟสเป็น 0โวลต์ด้วยโดยไม่ได้ต่อลงกราวด์โดยตรงจึงถูกเรียกว่า กราวด์เทียม (Firtual ground)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 10.2 วงจรดีทูเอแบบใช้ตัวต้านทานหลายค่า

มาดูกันตอนที่สวิตซ์ D_0 ตัวต้านทาน R_1 ค่า 100 กิโลโอห์มจะมีแรงดัน 5 โวลต์ที่ปลายข้างหนึ่ง อีกข้างหนึ่งเป็น O โวลต์ (กราวด์เทียม) 5 โวลต์ ของโอห์มจะมีแรงดันตกคร่อม 5 โวลต์ ซึ่งให้กระแสไหลผ่าน 0.05 มิลลิแอมป์ กระแสนี้ไม่อาจเข้าไปยังอินพุตของออปแอมป์ได้ เนื่องจากออปแอมป์มีความต้านทานอินพุตสูงมาก และไม่สามารถส่งหรือรับกระแสมากๆได้ ดังนั้น กระแส 0.05 mA จึงไม่ไหลผ่านไปยังเอาต์พุต โดยผ่านตัวต้านทานป้อนกลับ Rf 10 กิโลโอหม์ จะได้แรงดันเอาต์พุต เท่ากับ (10 กิโลโอห์ม)* (-0.05 มิลลิแอมป์) = -0.05โวลต์ เพื่อรับกระแสผ่านสวิทช์ D_0 และเพื่อรักษาสภาวะกราวด์เทียมไว้ แต่ถ้าหากยังสงสัยในสภาวะกราวด์เทียม ก็ลองวาดเป็นวงจรดิไวเดอร์ที่มีแรงดันข้างหนึ่ง +5 โวลต์ตรงกลางเป็น 0 โวลต์ และอีกปลายหนึ่งมีค่า -0.5 โวลต์

เมื่อเปิดวงจรที่สวิทช์ D_0 และปิดวงจรที่สวิทช์ (ขณะที่ R2 มีค่าเป็นครึ่งหนึ่งของ R1) กระแสเพิ่มเป็น 2 เท่า หรือ 0.1 มิลลิแอมป์ ไหลผ่าน R_f กราวด์เทียม และ R2 ทำให้มีแรงดันเอาต์พุต -1 โวลต์ ต่อไปก็ปิดวงจรทั้งที่ Do และ D_1 จะได้กระแส 0.05 มิลลิแอมป์ ไหลผ่าน R1 และ 0.1 มิลิแอมป์ผ่าน R2 รวมกระแส 0.15 มิลิแอมป์ ได้แรงดันเอาต์พุต -1.5 โวลต์

เมื่อเปลี่ยนการปิดเปิดสวิตซ์ไปเรื่อย ๆ จะได้แรงดันเอาต์พุตค่าต่าง ๆ กันกระแสที่ผ่านสวิตซ์แต่ละตัวจะถูกรวมกันที่จุดกราวด์เทียม แล้วเปลี่ยนเป็นแรงดันที่เอาท์พุต โดยตัวต้านทานในป้อนกลับ R_f

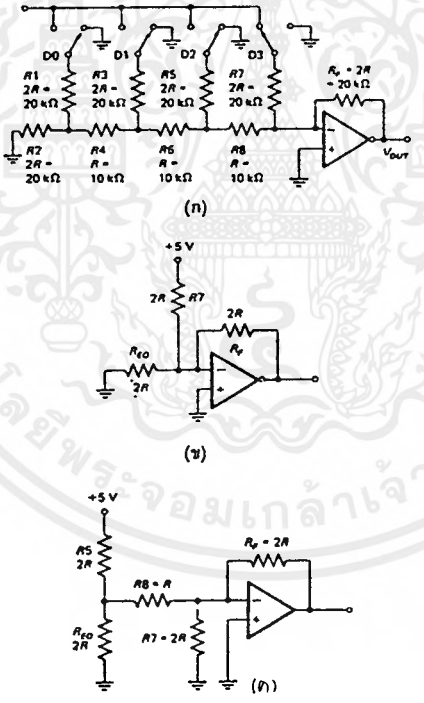
แรงดันเอาต์พุตจะเพิ่มขึ้นเป็นระดับ ๆ เหมือนขั้นบันได ดังนั้น 4 บิตจึงได้ 15 ระดับ แต่ละระดับต่างกัน - 0.5 โวลต์อาจกำหนดระยะห่างของแต่ละระดับได้โดยเปลี่ยนขนาดของ R_f มากเกินไป ออป- แอมป์ถึงจุดอิ่มตัว (ที่แรงดัน -14 โวลต์)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แบบใช้ตัวต้านทาน 2 ค่า

เมื่อวงจรดีทูเอ มีขนาดมากกว่า 4 บิต วงจรตามรูปที่ 10.4 จะเกิดปัญหาเนื่องจากต้องการค่าความต้านทานที่มีช่วงกว้างมาก วิธีที่ใช้หลังจากไบนารี่เวดเหมือนกัน แต่ใช้ความต้านทานเพียง 2 ค่า แสดงในรูปที่ 3 (ก) ซึ่งกระแสจะถูกเปลี่ยนค่าแรงดันโดยออปแอมป์และตัวต้านทานป้อนกลับ R_f เหมือนวงจรในรูปที่ 2วิธีนี้เรียกว่าการใช้ความต้านทาน 2 ค่า

สังเกตุให้ดี หลักการความต้านทาน 2 ค่า ดูไปก็คล้ายกับกฎของเดอร์ซ๊อฟ เพียงแต่ว่าค่าความต้านทานที่ใช้เป็นอัตราส่วนที่ทำให้คำนวณได้ง่าย แรกเลยสมมุติว่า สวิตซ์ D_3 ซึ่งเป็นสวิตซ์ในบิตที่มีนัยสำคัญสูงสุดดังนั้นต่อกับแรงดันอ้างอิง 5 โวลต์ ในขณะที่สวิตซ์ตัวอื่นปิดลงกราวด์ ดังนั้น R1 และ R2 จึงต่อขนานกันลงกราวด์ สังเกตตัวต้านทาน 2R ต่อขนานกับ 2R อีกตัวหนึ่งจึง มีค่าเท่ากับ R ค่า R นี้จะถูกบวกกัน R4 กลายเป็นค่า 2R แล้วขนานกันกับ R3 ลงกราวด์ การรวมของ R3 และตัวต้านทานก่อนหน้าจึงทำให้เหลือเพียงค่า R ต่ออนุกรมกับ R6 พิจารณาเช่น เดียวกันกับวงจรส่วนที่เหลือก็จะได้เป็นวงจรง่ายขึ้น ดังรูปที่ 10.3 (ข)



รูปที่ 10. 3 วงจรเปลี่ยนสัญญาณแบบ R/2R แลดเดอร์

(ก) วงจรสมบูรณ์

(ข) วงจรเหมือนในขณะที่สวิทซ์ของบิตที่มีนัยสำคัญสูงสุดปิด

(ค) วงจรเสมือนในขณะที่สวิทซ์ของบิตที่มีนัยสำคัญรองลงมาปิด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยเหตุที่กราวด์เทียมของออปแอมป์มีแรงดัน 0 โวลต์ ทำให้ไม่มีกระแสไหลผ่านค่าความต้านทานเหล่านี้ลงกราวด์ จึงไม่ต้องสนใจส่วนนี้ ดังนั้น แรงดัน 5 โวลต์ ที่ปลายข้างหนึ่ง ของ R7 ค่า 20 กิโลโอห์ม ทำให้มีกระแส 0.25 มิลิแอมป์ ผ่านที่จุดต่อและผ่าน Rf 20 กิโลโอห์ม แรงดันเาต์พุตที่ได้จากบิตที่มีนัยสำคัญสูงสุดจึงมีค่า -5 โวลต์ แรงดันที่ได้จากบิตที่มีนัยสำคัญรองลงมาก็หาได้โดยเปิดสวิตซ์ D2 ไปยัง +5 โวลต์และ D3 ลงกราวด์ ตัวต้านทานทั้งหมดที่อยู่ทางซ้ายของ R5 ในรูปที่ 3 (ก) ลดรูปลงเหลือเพียง 2R ต่อลงกราวด์

การวิเคราะห์วงจรสามารถนำทฤษฎีของเทวินินมาใช้ได้ โดยมีการแบ่งแรงดันระหว่าง R5 และ 2R ที่ต่อลงกราวด์ ดังรูป3 (ค) แรงดันของเทวินิน คือ แรงดันที่รอยต่อหรือ 2.5 โวลต์ ตัวต้านทานเทวินินมีค่าเท่ากับตัวต้านทาน 2 ตัวต่อขนานกัน (หรือ R) อุปกรณ์ที่อยู่ทางด้านซ้ายของ R6 สามารถลดรูปได้เหลือค่า R ต่อกับ 2.5 โวลต์ เราสามารถละทิ้ง R7 ได้เพราะจุดปลายทั้งสองค่าต่อลงกราวด์ค่าความต้านทานรวมระหว่างที่จุดรวม (จุดกราวด์เทียม) และแรงดันเทวินินคือ 2R หรือ 20 กิโลโอห์ม กระแสที่จุดรวมคือ 2.5 โวลต์ หรือ 0.125 มิลิแอมป์กระแสที่ผ่าน Rf = 20 กิโลโอห์ม ทำให้เกิดแรงดันเอาต์พุต -2.5 โวลต์ (สำหรับบิตที่มีนัยสำคัญถัดมา)

ด้วยการวิเคราะห์ในทำนองเดียวกันนี้ สามารถหาแรงดันเอาต์พุตที่บิตต่ำลงมาอีกได้ 1.25 โวลต์ และที่ค่าดิจิตอลต่ำสุดได้ 0.625 โวลต์ ในขณะที่ค่าดิจิตอลสูงสุด (สวิทซ์ทุกตัวต่อไปที่+5 โวลต์) ได้เอาต์พุตเต็มสเกลคือ 9.375 โวลต์

แม้ว่าดีทูเอ คอนเวอร์เตอร์ แบบ R/2R แลดเดอร์ จะวิเคราะห์ยากกว่าแบบใช้ตัวต้านทานหลายค่า(Weighted resistor) แต่จะง่ายกว่าสำหรับการต่อวงจรให้ถูกต้อง เพราะใช้ค่าความต้านทางเพียง 2 ค่าเท่านั้น จำนวนบิตก็เพิ่มได้โดยเพิ่มส่วนของ R/2R วงจรให้ถูกต้องเพราะ ใช้ค่าความต้านทานเพียง 2 ค่าเท่านั้น จำนวนบิตก็เพิ่มได้โดยเพิ่มส่วนของ R/2R ลงไป วงจรนับ 4 บิตที่เป็น TTL หรือ CMOS อาจนำมาต่อแทนตำแหน่งของสวิทซ์ในวงจรรูปที่10.5 (ก) เพื่อให้แรงดันเอาท์พุตเป็นขั้นบันไดได้

แบบใช้ไอซี

โมโนลิทิค (Monolithic) หมายถึง หินก้อนเดียว เมื่อนำมาใช้ในวงจรรวมจะเป็นการชี้บอกว่าวงจร ๆ หนึ่งถูกบรรจุอยู่บนสารกึ่งตัวนำเพียงชิ้นเดียว ส่วน ไฮบริด (hybrid) บรรจุสารกึ่งตัวนำที่เรียกว่าชิพ (chip) เพียงชิ้นเดียวหรือมากกว่า มีตัวต้านทานหรือตัวประกอบวงจรอื่น ๆ อยู่ในกรอบของไอซีตัวเดียว

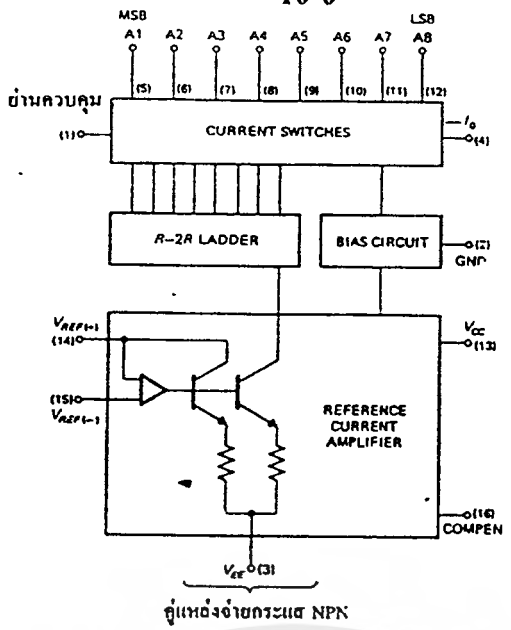
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างวงจรเปลี่ยนสัญญาณดีทูเอโมโนลิทิคขนาด 8 บิต คือ MC 1408 ซึ่งมีผังการทำงานดังแสดงในรูปที่ 4 (ก) 1408L เป็น DIP (Dual Inline Package) 16 ขา ใช้ Vcc +5 โวลต์ และ Vee จาก -5 โวลต์ (ต่ำที่สุด) ถึง -15 โวลต์ (สูงสุด) ใน 1408L R/2R แลดเดอร์ แบ่งกระแสที่ได้จากภาคขยายเป็น 8 ระดับขึ้นอยู่กับค่าทางเลขฐานสอง (binary) ทรานซิสเตอร์แบบไบโพลาร์จะสวิตซ์ให้กระแสที่ได้สอดคล้องกับอินพุต A1 ถึง A8 การเรียงจากบิตที่มีนัยสำคัญสูงสุดถึงบิตที่มีนัยสำคัญต่ำสุดจะกลับกันกับของวงจรนับทั่ว ๆ ไป แต่วงจรเปลี่ยนสัญญาณดิจิตอลเป็นอนาลอกของตัวก็จะไม่ได้เรียงอย่างนี้ ดังนั้น ควรอ่านคู่มืออย่างละเอียดถี่ถ้วนเสียก่อน

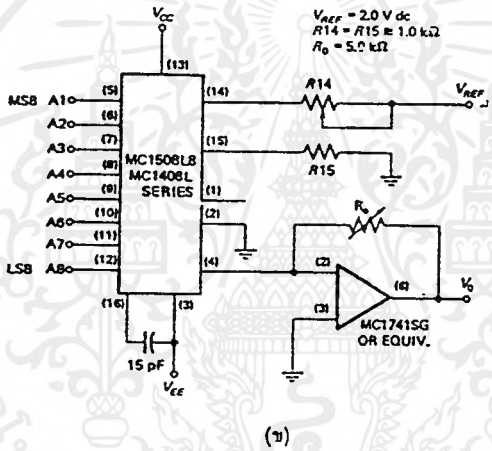
1408L มีกระแสเอาต์พุตที่สามารถเปลี่ยนเป็นแรงดันได้ด้วยออปแอมป์และตัวต้านทานดังแสดงในรูปที่ 4 (ข) แรงดันนี้สามารถคำนวณโดยใช้สูตร

$$V_{out} = \frac{V_{ref}}{R_{14}} * \left(\frac{A1}{2} + \frac{A2}{4} + \frac{A3}{8} + \frac{A4}{16} + \frac{A5}{32} + \frac{A6}{64} + \frac{A7}{128} + \frac{A8}{256}\right)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



(ก)



(ข)

รูปที่ 10.4 วงจรดีทูเอ MC1408 ของโมโตโรลา

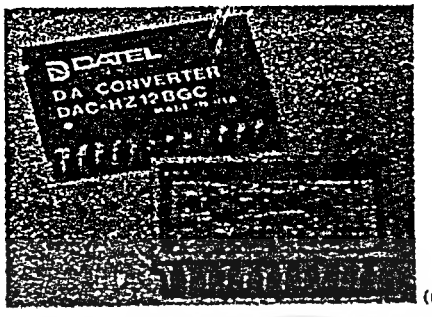
(ก) บล็อกไดอะแกรม

(ข) การต่อแรงดันเอาต์พุต

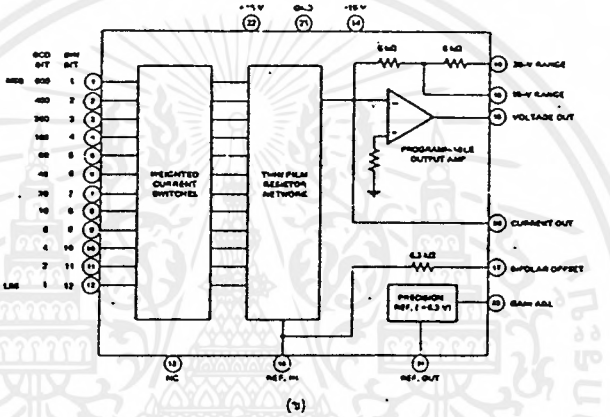
จากค่าที่เลือกไว้ ได้แรงดันเอาต์พุตเต็มสเกล (อินพุต A1 ถึง A6 เป็น "1") คือ

$$2 \text{ โวลต์} * \frac{5 \text{กิโลโอห์ม}}{1 \text{กิโลโอห์ม}} * \frac{255}{256}$$

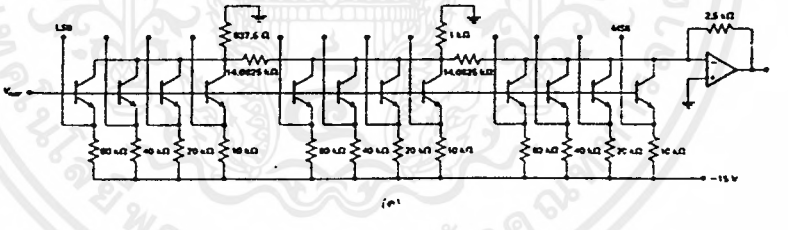
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



(ก)



(ข)



(ค)

รูปที่ 10.5 วงจรเปลี่ยนสัญญาณดีทูเอแบบไฮบริดจ์ของ Datel

(ก) ลักษณะภายนอก

(ข) บล็อกไดอะแกรม

(ค) แหล่งจ่ายกระแสแบบเรียงค่าไบนารี่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซึ่งเราถือว่าเป็นวงจรเปลี่ยนสัญญาณแบบ 10 โวลต์ เต็มสเกล ตัวอย่าง การนำไปใช้งาน ของ 1408L เช่น วงจรกำเนิดเสียง โดยรูปคลื่นเอาต์พุต ของ วงจรนับ 8 บิต สามารถนำมาต่อกับดิน พุตของดีทูเอ ซึ่งประกอบขึ้นด้วยขั้นบันไดเล็ก ๆ ถึง 255 ขั้น ความถี่เอาต์พุต เท่ากับความถี่ของ สัญญาณนาฬิกาทางอินพุต หารด้วย 256 คลื่นรูปอื่น ๆ ก็อาจทำได้โดยต่อดีทูเอกับเอาต์พุตของ หน่วยความจำ ROM หรือRAM แบบ 8 บิต หน่วยความจำได้ถูก โปรแกรม ด้วยค่าเลขฐานสองค่า ต่าง ๆ ดังนั้น เสียงที่ได้จะมีความหลากหลาย ฟังดูแปลกหู หรือ อาจโปรแกรมเสียงตามที่ต้องการก็ ได้ ข้อจำกัดของเสียงที่ออกมาก็ขึ้นอยู่กับจินตนาการของผู้ที่โปรแกรม จำนวนหน่วยความจำ และความสามารถในการโปรแกรมเอง

ถ้าต้องการใช้งานที่บิตมากกว่านี้ ก็อาจเปลี่ยนเป็นของบริษัท Datel เบอร์ DAC-HZ 12 BGC ซึ่งมี 24 ขา รูปร่างได้แสดงไว้แล้วในรูปที่ 5 (ก) และผังการทำงานได้แสดงไว้ใน รูปที่ 5 (ข) เป็น วงจรเปลี่ยนสัญญาณ 12 บิต ต้องการไฟเลี้ยง+15 โวลต์ และ -15 โวลต์ มีตัวต้านทานและออป แอมป์รวมไว้ในตัวแล้วโดยจะให้ เอาต์พุตออกมาเป็นกระแสหรือ แรงดันก็ได้แรงดันเต็มสเกล สามารถเปลี่ยนได้โดยการเปลี่ยนค่าตัวต้านทานที่มีช่วงกว้างมากหากเกินกว่า 4 บิต ปัญหานี้อาจ แก้ไขได้โดยใช้ขนาด 4 บิต 3 ชุดต่อกันดังรูปที่ 5 (ค)

ลักษณะสมบัติของดีทูเอ

ลักษณะสมบัติอันแรก ของการแปลงสัญญาณ ดิจิตอล เป็นอะนาล็อก ที่จะพูดถึงคือความ ละเอียด(resolution) ซึ่งขึ้นกับจำนวนของบิตทางด้านอินพุตตัวอย่าง เช่น วงจรเปลี่ยนสัญญาณ 8 บิต มีระดับเอาต์พุต 2^8 หรือ 256ระดับดังนั้นความละเอียดคือ 1 ใน 256 วงจรเปลี่ยนสัญญาณ 12 บิต มีความละเอียด 2^{12} หรือ 4096 ความละเอียดบางครั้ง จะคิดเป็นเปอร์เซ็นต์ คือ $1/4096 = 0.024\%$

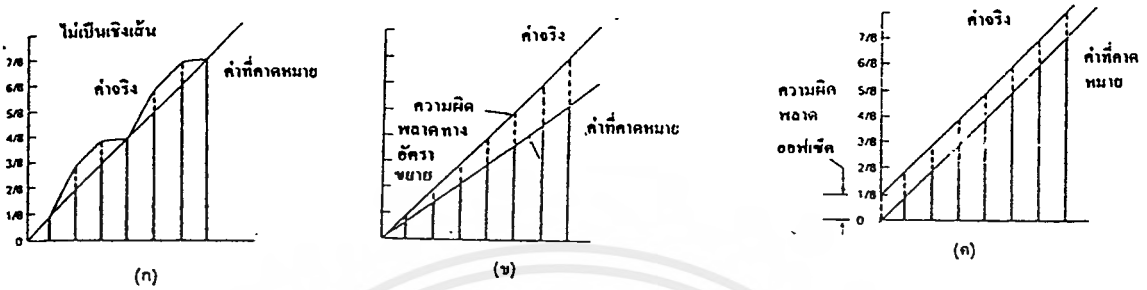
ลักษณะสมบัติข้อต่อมาคือ ความถูกต้อง (accuracy) ดีทูเอ ความถูกต้องจากการเปรียบ เทียบระหว่างเอาต์พุตจริงและเอาต์พุตที่ปรากฏ โดยคิดที่เต็มสเกล ถ้าวงจรเปลี่ยนสัญญาณมีเอาต์ พุตเต็มสเกล 10 โวลต์ มีความถูกต้อง$\pm$0.2% ดังนั้นความผิดพลาดสูงสุดคือ 0.002 * 10 โวลต์ หรือ 20 มิลลิโวลต์ ในทางทฤษฎีแล้ว ความถูกต้องของวงจรเปลี่ยนสัญญาณดิจิตอลเป็น อนาลอกไม่ ควรต่ำกว่า $\pm 1/2$ ของค่าที่ LSR (บิตที่มีนัยสำคัญต่ำสุด)

วงจรเปลี่ยนสัญญาณ 10 บิตมีความละเอียด 1/1024 หรือประมาณ 0.1% ความถูกต้องควรมีค่า $\pm$0.05

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ความผิดพลาดอาจเกิดขึ้นได้หลายประการ ในรูปที่ 10.6 แสดงไว้ 3 แบบ ดังนี้ คือ

(1) ความผิดพลาดเชิงเส้น (Linearity errors) ค่าจริงที่ได้จากเอาต์พุตจริงต่างจาก เอาต์พุตตามทฤษฎีที่ควรจะเป็นเส้นตรง ความผิดพลาดนี้มักจะมาจากความผิดพลาดจากแหล่งจ่ายกระแส หรือค่าความต้านทาน



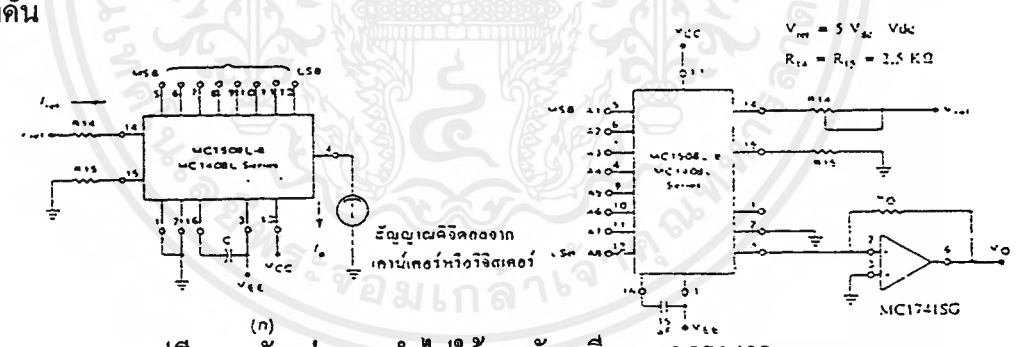
รูปที่ 10.6 ความผิดพลาดที่เกิดขึ้นในการเปลี่ยนสัญญาณ

(ก) เชิงเส้น

(ข) อัตราขยาย

(ค) ออฟเซต

ความผิดพลาดแบบที่สอง คือ ความผิดพลาดทางอัตราขยาย (gain error) ความผิดพลาดนี้มักเกิดจากความผิดพลาดของตัวต้านทาน ป้อนกลับของออปแอมป์ที่ทำหน้าที่เปลี่ยนจากกระแสเป็นแรงดัน



รูปที่ 10.7 ตัวอย่างการนำไปใช้งานอันหนึ่งของ MC1408

(ก) มิเตอร์วัดกระแส

(ข) ดีทูเอคอนเวอร์เตอร์

ความผิดพลาดออฟเซต หรือ offset error คือ เมื่ออินพุตทุกตัวเป็นศูนย์แล้ว เอาต์พุตไม่เป็นศูนย์ ทำให้เอาต์พุตมีค่าแรงดันผิดพลาดค่าหนึ่งบวกกับค่าจริงอยู่ตลอดเวลา ดังในรูปที่ 10.6 (ค) ความผิดพลาดนี้เกิดจากความผิดพลาด ของการขยายของออปแอมป์ และกระแสรั่วไหลที่กราวด์สวิตช์

ลักษณะคุณสมบัติต่อมาคือ โมโนโทนิค (monotonicity) จะเรียกว่าเป็น โมโนโทนิคก็ต่อเมื่อไม่มีการกระโดดข้ามขั้นตลอดย่านการใช้งาน ส่วนเวลาเซตเอาต์พุต (output setting time)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

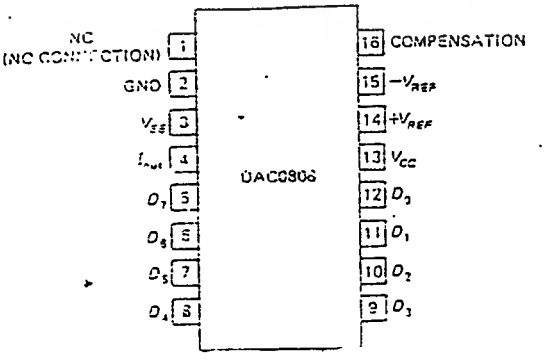
เป็นเวลาที่เอาต์พุต ของวงจรเปลี่ยนสัญญาณที่ใช้ในการเพิ่มขึ้นถึง ± 1/2 ของ LSB หลังจากมีการเปลี่ยนแปลงทางอินพุต ถ้าวงจรเปลี่ยนสัญญาณถูกใช้งานย่านความถี่สูง อาจทำให้มีการเพิ่มแรงดันไม่ถึงค่าที่ถูกต้องทำให้เกิดความผิดพลาดขึ้นได้อีกประการหนึ่ง

วงจรรวมทาง D/A

มีไอซีตัวแปลงสัญญาณ D/A ต่าง ๆ มากมายที่ใช้งาน โดยทั่วไปมีอินพุตตั้งแต่ 8 ถึง 18 บิต แต่การใช้งานส่วยมากจะใช้ 8 ถึง 12 บิต มีแหล่งกำเนิดกระแสมากมายที่ถูกสร้างขึ้นเพื่อป้อนให้แก่ไปนารี่แล็ตเดอร์ เหตุที่พวกเราใช้ประโยชน์จากกระแสมากกว่าแรงดัน เพราะว่าไบนารี่แล็ตเตอร์สำคัญพอ ๆ กับกระแสคงที่ เพราะทำให้วงจรทางคาปาซิแตนซ์จะส่งผลกระทบต่อวงจรน้อยมากและทำให้การทำงานมีความเร็วขึ้น

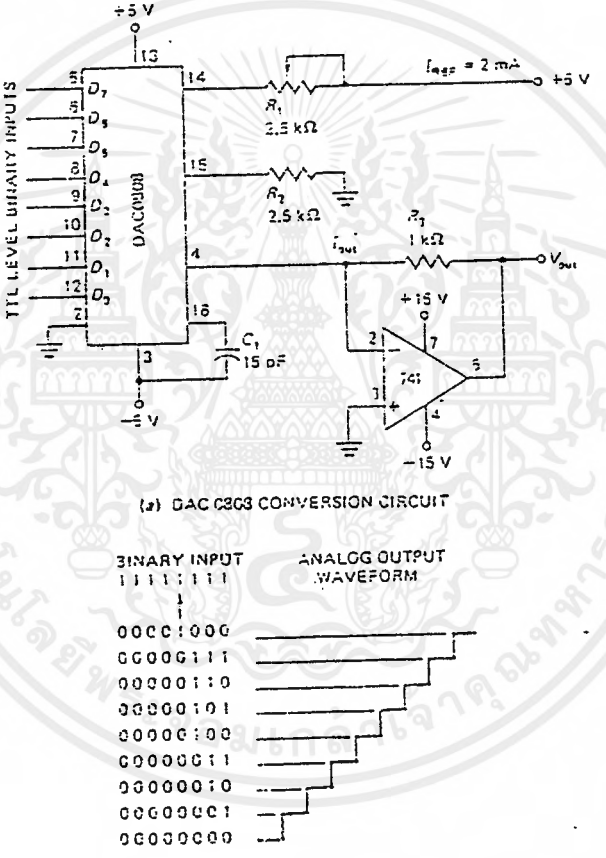
IC DAC0808

DAC0808 เป็นอุปกรณ์ตัวแปลง D/A ที่มีราคาถูกตัวหนึ่ง ซึ่งมีแหล่งกำเนิดกระแสอ้างอิงภายในตัวมันเอง DAC0808 มีลักษณะดังในรูปที่ 5 โดยทั่วไป ขา 3 มีค่าเท่ากับ -15 V ขา 4 เป็นกระแสแล็ตเตอร์ และตามปกติจะต่อกับออปแอมป์ขา 5 ถึง 12 เป็นอินพุตของข้อมูลทางดิจิตอลขนาด 8 บิต ขา 13 เท่ากับ +5 V ขา 14 จะถูกต่อกับความต้านทานเพื่อจำกัดกระแสให้เอาท์พุตออกมาเป็นแรงดันค่าบวก ขา 15 ถูกต่อลงกราวด์ผ่านความต้านทานตัวหนึ่ง และขา 3 ถึง 16 จะต่อกับคาปาซิเตอร์ตัวหนึ่งเพื่อชดเชยความถี่และป้องกันอุปกรณ์จากการออสซิเลต



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 10.8 แสดง DAC0808 ต่อเป็นตัวแปลงสัญญาณ D/A ตัวต้านทาน R□1□ คือ ความต้านทานปรับค่าได้เพื่อเซตค่ากระแสให้เป็น 2 ตัวต้านทาน R 2 มีค่าเท่ากับตัวต้านทาน R1 ต่อเพื่อชดเชยกระแสในอินพุตของตัวแปลงสัญญาณ D/A กระแสเอาท์พุตที่ขา 4 ใช้ในการขับอินเวอร์ดติ้งออปแอมป์ให้ทำงาน จนกระทั่งแรงดันเอาท์พุตของออปแอมป์แปรตามกระแสเอาท์พุตของตัวแปลงสัญญาณ D/A จาก 0 ถึง 1.992 V



รูปที่ 10.8 ความเที่ยงตรงและค่าความละเอียด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไอซี DAC0808 มีความเที่ยวตรงเท่ากับ $\pm$1/2 LSB โดยความเที่ยงตรงนี้เรา หมายถึง ว่าระดับเอาท์พุตมีค่าเป็นสัดส่วนใกล้เคียงกับระดับเอาท์พุตเต็มสเกลเท่าไร จากตัวอย่างอุปกรณ์ DAC0808 ขนาด 4 บิต มีระดับเอาท์พุตตามอุดมคติแสดงค่าเป็นสัดส่วนกับระดับเอาท์พุตเต็มสเกลเป็น 0,1/15,2/15,3/15 ฯลฯ

ค่าความเที่ยวตรงของตัวแปลงสัญญาณ D/A ถูกกำหนดโดยจำนวนค่าที่เพิ่มขึ้นทางเอาท์พุตจะได้

ค่าที่เพิ่มขึ้น(เป็นสเตป) = 2^{n-1}

ดังนั้นยิ่งจำนวนทางเอาท์พุตยิ่งมาก ค่าทางสเตปจะยิ่งมาก และค่าความละเอียดจะเพิ่มขึ้นด้วย แต่ในความเป็นจริงแล้ว ค่าความละเอียดจะแสดงให้เห็นถึงจำนวนบิตทางอินพุต จากตัวอย่างถ้าอุปกรณ์อินพุตขนาด 8 บิต จะกล่าวว่ามีค่าความละเอียดเป็น 8 บิตด้วย

เซททริ้งทามส์

เมื่อเรานำตัวแปลงสัญญาณD/A มาใช้งานทางด้านเวลา เพื่อให้เอาท์พุตอยู่ในสถานะเสถียรภาพหลังจากที่มีสัญญาณอินพุตทางดิจิตอลปรากฏขึ้น โดยเซททริ้งทามส์ คือ เวลาที่ต้องการเพื่อให้ตัวแปลงสัญญาณ D/A อยู่ในสถานะเสถียรภาพภายใน $\pm$1/2LSB ของเอาท์พุตชุดสุดท้ายการเซททริ้งทามส์สำหรับตัวแปลงสัญญาณ D/A มีอยู่ในตารางข้อมูลของอุปกรณ์นั้นๆ และมันมีความสำคัญเพราะเป็นตัวจำกัดความเร็วที่อุปกรณ์ตัวนั้นใช้ประโยชน์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทสรุป

จากบทความที่กล่าวมาทั้งหมด ได้แสดงถึงกระบวนการต่าง ๆ ที่เกิดขึ้นของตัวแปลงสัญญาณ D/A ตั้งแต่การทำงาน การส่งข้อมูล การกำหนดค่าสถานะของขา ความแน่นอนและค่าความละเอียดที่เกิดขึ้น ฯลฯ ดังนั้น เราจึงสรุปหลักการต่าง ๆ ที่อธิบายมาทั้งหมดแยกเป็นหัวข้อต่าง ๆ ดังนี้

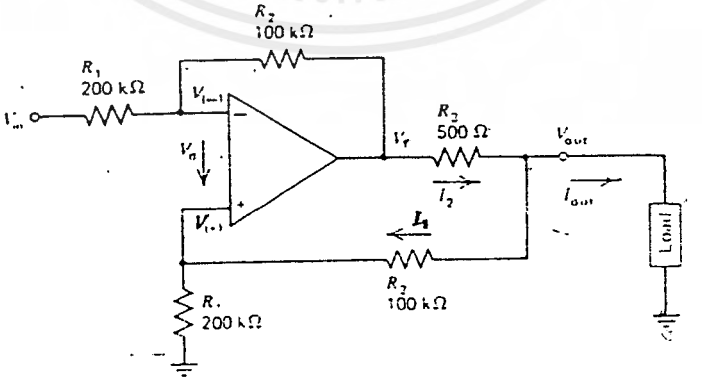
1. ตัวแปลงสัญญาณ D/A จะเปลี่ยนสัญญาณอินพุตทางดิจิตอล เป็นแรงดันหรือกระแส
2. ตัวแปลงสัญญาณ D/A จะใช้งานอยู่ที่อินพุตตั้งแต่ 8 ถึง 18 บิต
3. การแปลงสัญญาณแบบ D/A ง่ายกว่าการแปลงสัญญาณแบบ A/D
4. ข่ายวงจรความต้านทานจะถูกใช้ประโยชน์สำหรับการแปลงแบบ D/A บ่อยมาก
5. ข่ายวงจรความต้านทาน D/Aที่มีประสิทธิภาพมากที่สุด คือไบนารี่แล็ตเตอร์ หรือ ข่ายวงจรความต้านทาน R-2R
6. กระแสจะคงที่ในข่ายวงจรแบบ R-2R ซึ่งทำให้แรงดันคงที่ด้วย และ ทำให้คาปาซิแตนซ์ส่ง ผลกระทบน้อยมากในกระบวนการ D/A
7. เซททริ้งทามส์จะถูกกำหนดโดยความเร็ว ที่ใช้เพื่อให้กระบวนการ D/A เสร็จสมบูรณ์ มันเป็นเวลาที่ต้องการให้ เอาท์พุตเข้าถึงการเสถียรภาพภายในช่วง $\pm$1/2 LSB ของค่าสุดท้าย
8. ไอซี DAC0808 มีกระแสอ้างอิงที่ถูกสร้างขึ้นภายในตัวมันเอง
9. ความแม่นยำของ D/A จะเป็นสมการของเอาท์พุตที่แท้จริงกับเอาท์พุตทางอุดมคติว่ามีค่าใกล้เคียงกันเท่าไร
10. ค่าความละเอียดของ D/A จะเป็นสถานะที่แสดงถึงจำนวนของบิตทางอินพุต เช่น ค่าความละเอียดขนาด 8 บิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

10.2 VOLTAGE TOCURRENT CONVERTER

เนื่องจากการส่งสัญญาณเป็นแรงดันมีปัญหาเกิดขึ้นมากมาย ซึ่งทำให้เกิด %ERROR ขึ้นได้ กระแสนั้นมีบทบาทสำคัญมากในการส่งสัญญาณเป็นแบบ LOO การเปลี่ยนสัญญาณเป็นกระแส และการส่งสัญญาณไปยังโหลด นั้น จะไม่มีการสูญเสียของสัญญาณ เนื่องจากค่าความต้านทาน ของสายไฟ หรือ จุดต่อที่ไม่ดีในการส่งสัญญาณให้กับ กระบวนการมาตรฐานที่ใช้ในการส่ง สัญญาณ คือ ถ้าเป็นความดัน จะมีค่า 3-15 PSI แรงดันไฟฟ้าใช้ 1-5 VOLT และถ้าเป็นกระแสจะ ใช้ 4-20 mA ในการส่งสัญญาณ 0-100 % สาเหตุที่ 0% สัญญาณที่ส่งมีค่าไม่เท่ากับ 0 นั้น เพราะว่า จะได้ทำให้ทราบว่ามีสัญญาณควบคุมอยู่หรือไม่ โดยแสดงค่า3 PSI,1 VOLT และ 4 mA ให้เราได้ ทราบ การเลือก วงจร V TO I จะขึ้นอยู่กับ ค่าความต้านทานของโหลดและการต่อกราวด์ของ โหลดแบบ FLOATION LOAD จะช่วยลดสัญญาณรบกวน และอีกแบบหนึ่งคือ GROUNDLOAD ซึ่งก็เหมาะกับการใช้ในการขับสัญญาณเพราะ ส่วนแสดงผลและอุปกรณ์อิเลคทรอนิคส์ที่ใช้ใน การควบคุม ส่วนใหญ่จะออกแบบมาในลักษณะเดียวกับแบบ GROUND LOAD

วงจรอย่างง่ายสำหรับ V/I ดังแสดงในรูปซึ่งเป็น NON INVERTING AMPLIFIER CIRCUIT การส่งสัญญาณและการควบคุม LOAD นี้ได้จากสัญญาณที่ขับจาก ส่วนป้อนกลับ แบบลบซึ่งเราสามารถวิเคราะห์วงจรง่าย ๆ คือ OP-AMP เป็นตัวทำหน้าที่ต่อครบ LOOP และ แรงดันที่ขา NON INVERTING จะปรากฏที่ขา NON INVERING ด้วย แต่แรงดันนี้จะตกคร่อม ความต้านทานอยู่ดังนั้น กระแสที่ไหลผ่าน R คือ I



รูปที่ 10.9 การเปลี่ยนกระแสเป็นแรงดันอย่างง่าย

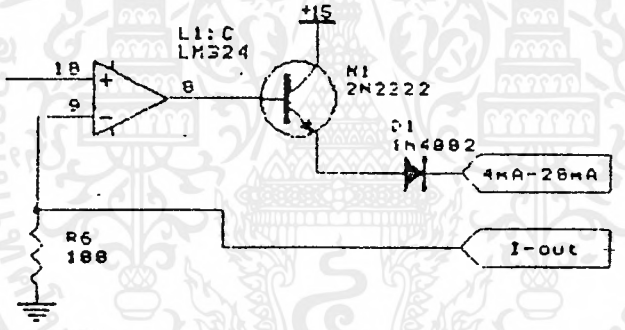
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วงจรการเปลี่ยนกระแส เป็นแรงดันง่าย ๆ มีข้อพิจารณาหลายข้อที่ต้อง พิจารณาในการใช้วงจร V/I แบบโหลดลอย ดังวงจรในรูป

ค่าความต้านทานในรูปที่ใช้ในการส่ง (R loop = R wire + R load) จะไม่มีผลต่อกระแสในการส่ง แต่แรงดันเอาท์พุทจะขึ้นอยู่กับ R loop

$$V_{out} = (I+R_{loop}/R)\, E_{in} < V_{sat}$$

โดยต้องพยายามทำให้ R loop มีค่ามากเกินไป จนทำให้เกิดการ saturation ได้ Opamp ต้องสามารถจ่ายกระแสได้เพียงพอความต้องการ เพราะมีการส่งสัญญาณอยู่หลายมาตราฐานที่ต้องใช้กระแส 20-60 mA ซึ่ง OP-AMP ทั่วไปไม่สามารถจ่ายได้จึงต้องใช้ TRANSISTOR ช่วย ดังรูป



รูปที่ 10.10 วงจร V/I แบบมีตัวขับกระแส

เนื่องจาก Transistor อยู่ใน LOOP การป้อนกลับแบบลบ OP-AMPจึงสามารถทำการชดเชย TRANSISTOR ในเรื่องการ BIAS, การ OFFSET & NON- ถ้าสัญญาณต้องเปลี่ยนไปถึงช่วงล่าง ซึ่งมีกระแสไหลกลับทิศทางเราต้องใส่ QZ(PNP) ซึ่งเป็น COMPLEMENT ของ Q1 สำหรับจ่ายกระแสช่วงลบกระแสจากโหลด ต้องไหลกับมาตามที่สายที่ต่อกลับ OP-AMP เพราะวงจรไม่สามารถขับกระแสต่อลงกราวน์ได้เลย ซึ่งเป็นสัญญาณที่ต้องใช้สาย 2 เส้น ในการทำงานของวงจรซึ่งกระแสที่ไหลในสายทั้งสองมีทิศทางตรงกันข้ามกันเราสามารถใช้ความแตกต่าง หรือวงจรขยายเครื่องมือวัดที่โหลด เพื่อลดสัญญาณรบกวนที่สายสัญญาณทั้ง 2 เส้น ระหว่างการส่งสัญญาณควบคุมห้ามทำการเปิดโหลด เพราะ ถ้าถอดโหลดออกจะทำให้มีการป้อนกลับแบบลบซึ่งจะส่งผล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ให้ OP - AMP เกิดการ SATURATION โดยเราสามารถป้องกันได้โดยการ HOT OUTPUT ซึ่งจะทำให้วงจรมีการทำงานเหมือน วงจรตามแรงดันสัญญาณกระแสก็จะไม่มีผลต่อวงจร NON-INVERTION จะเป็นตัวป้องกันแหล่งจ่ายแรงดันอินพุทถูกโหลดถึงกระแส เราสามารถสร้าง วงจร I to V โดยใช้วงจรขยายแบบ INVERTING ได้

จากวงจรในรูปปัญหาที่จะเกิดขึ้นอีกอย่างหนึ่งก็คือเมื่อ

$$e_{in} = 0$$

$$I_L = 0$$

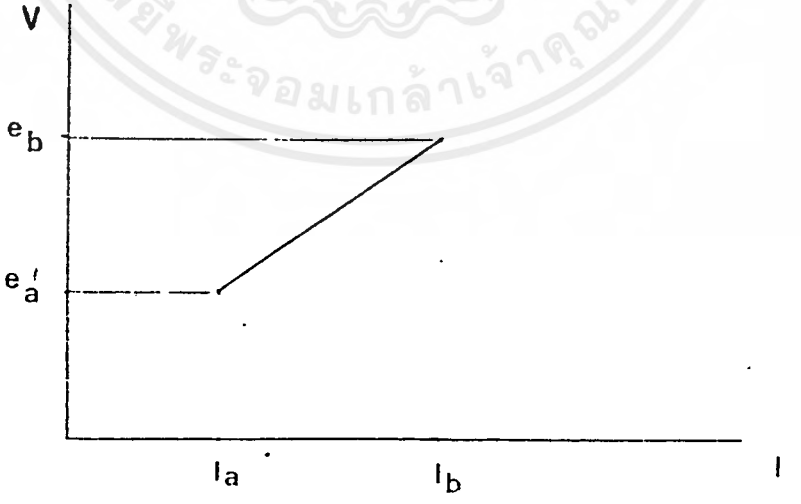
ในลักษณะเดียวกันถ้าเปิดโหลดหรือการส่งสัญญาณของ อุปกรณ์อิเลคทรอนิคส์ เกิดการผิดพลาด ก็จะทำให้ $I_L = 0$ เช่ กระแสจะตกลงเป็นศูนย์เมื่อ Ein = 0

เราสามารถแก้ปัญานี้ได้โดยการปรับ OFFSET ให้

$$e_{in} = 0 \text{ หรือ } e_{in} = \text{MINIMUM}$$

$$I_L = I(o)on$$

แรงดันอินพุทที่มีค่าแรงดันเกือบเป็นศูนย์ จะทำให้กระแสในลูปมีค่าไม่เท่า กับศูนย์วงจร OFF SET จะทำให้กระแสในสูปที่มีค่ามากกว่าหรือเท่ากันมีค่าเท่ากับ I(o) ถ้าวงจรเกิดการผิดพลาด

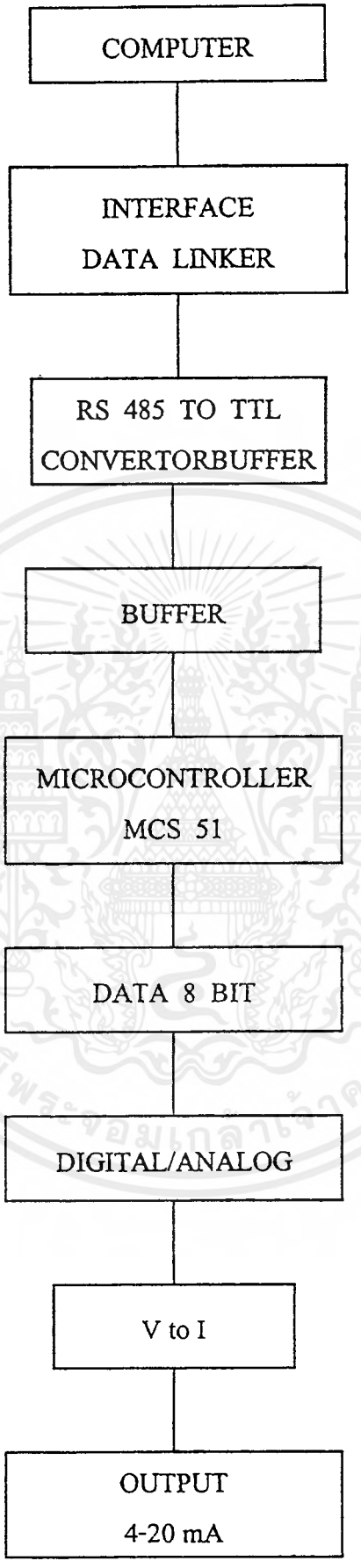


รูปที่ 10.11 วงจรและเส้นกราฟแสดงความสัมพันธ์ของ V/I CONVERTER

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วงจร V/I ที่มี OFFSET ดังแสดงในรูป โดยใช้วงจรขยาย NON - INVERTING แบบมี SUMMING แทนในวงจรรูป กระแสเอาท์พุทจะเปลี่ยนตามแรงดันอินพุท (e_{in}) และแรงดันอ้างอิง (e_{ref}) ค่าความต้านทานสองตัวจะป้องกันการโหลดของแหล่งจ่ายของแรงดันตัวอื่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



บล๊อคไดอะแกรมการทำงานของ ANALOG OUTPUT UNIT

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

<u>หน้าที่ บล็อกต่าง ๆ</u>

บล็อกที่ 1 มีหน้าที่ เป็นคอมพิวเตอร์

บล็อกที่ 2 มีหน้าที่ เป็นบล็อกของ Interface Data Linker จะทำหน้าที่รับข้อมูลที่เป็นสัญญาณมาตราฐาน RS-232 ให้เป็นสัญญาณในระบบมาตราฐาน RS-485

บล็อกที่ 3 มีหน้าที่ เป็นตัวที่จะรับข้อมูลที่คอมพิวเตอร์ส่งออกมาแล้วทำการที่ได้รับในระบบ RS-485 ให้เป็นระดับสัญญาณ TTL แล้วส่งให้ Micricontroller

บล็อกที่ 4 มีหน้าที่ เป็นบัฟเฟอร์ขนาด 8 บิท สำหรับการสื่อสารแบบอนุกรมทั้งการรับข้อมูลและการส่งข้อมูลโดยแบ่งเป็นบัฟเฟอร์ทางด้านการรับข้อมูล 1 ชุด 8 บิท และ บัฟเฟอร์ทางด้านการส่งข้อมูล 8 บิท โดยที่ CPU จะทำการจัดการเลือกบัฟเฟอร์โดยอัตโนมัติ

บล็อกที่ 5 มีหน้าที่ เป็น Main Board ที่จะใช้เป็นตัวคำนวณและแปลงค่าข้อมูลแบบ 1 บิท แบบอนุกรม แล้วจะส่งข้อมูลแบบขนานออกไปครั้งละ 8 บิทโดยจะใช้ Microcontroller เป็นตัวแปลงข้อมูล และจะใช้ Microcontroller เป็นตัวที่จะติดต่อสื่อสารกับคอมพิวเตอร์อีกด้วย

บล็อกที่ 6 มีหน้าที่ รับข้อมูลขนาด 8 บิทแบบขนานจาก Micricontroller แลจะทำการส่งไปภายนอก

บล็อกที่ 7 มีหน้าที่ เปลี่ยนสัญญาณจาก DIGTAIAL เป็น ANALOG โดยใช้ IC DAC 0808 และ ปรับ ให้ได้ 1- 5 V

บล็อกที่ 8 มีหน้าที่ เปลี่ยนแรงดันเป็นกระแส

บล็อกที่ 9 OUTPUT ที่ออกมาเป็นกระแส อยู่ในช่วง 4 -20 mA ใช้ควบคุมทรานสมิตเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้าที่หลักของ Analog Output Unit

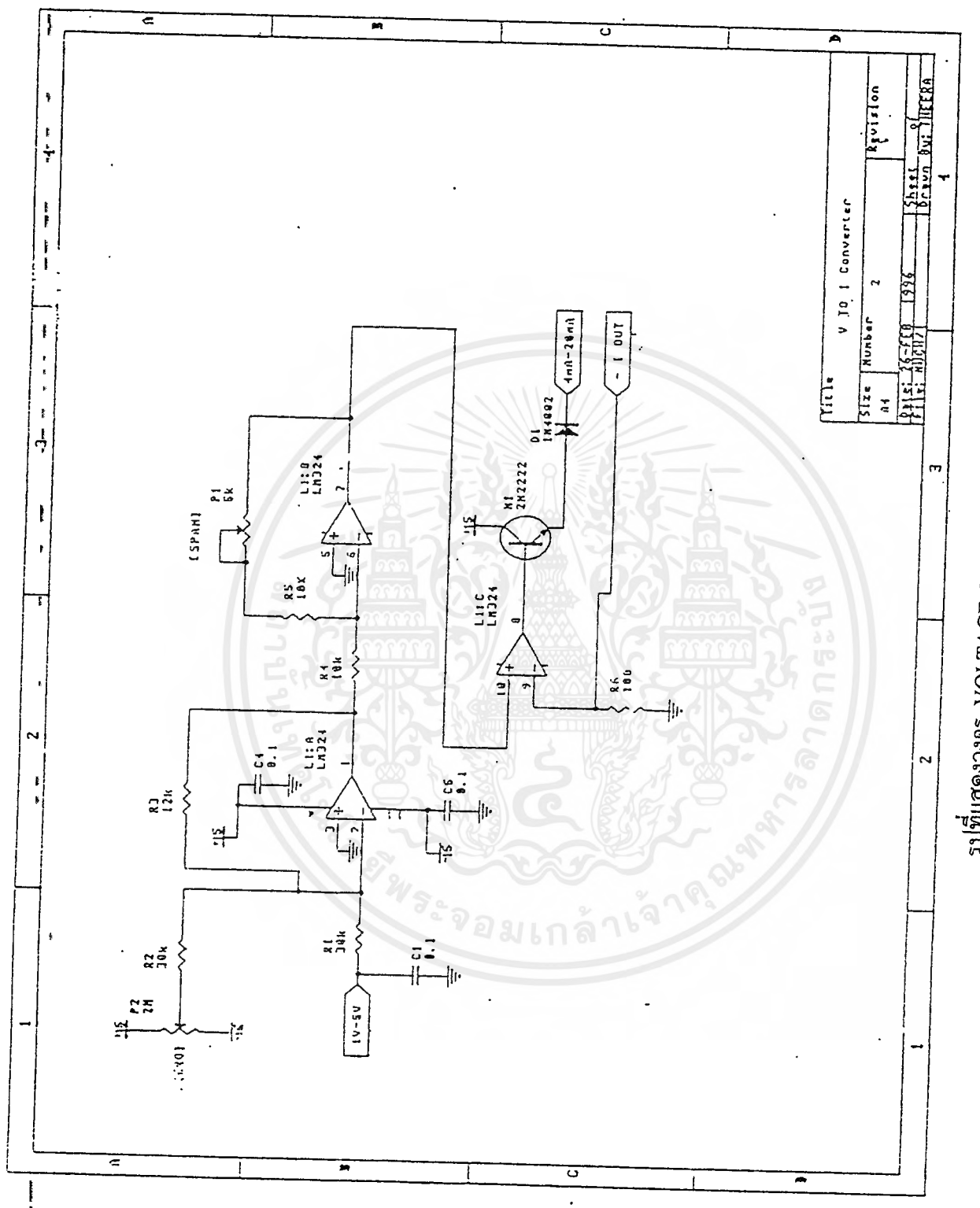
ภาค ANALOG OUTPUT UNIT เป็นส่วนหนึ่งในหลาย ๆ ส่วนของระบบ SMALL SCALE DCS โดยในส่วนของ ANALOG OUTPUT UNIT มีลักษณะการทำงานดังนี้

ANALOG OUTPUT UNIT จะได้รับข้อมูลจากสายสัญญาณเป็นการส่งในระดับ RS 485 ซึ่งลักษณะของข้อมูลที่ส่งมาตามสายสัญญาณจะเป็นข้อมูลแบบอนุกรม (SERIES DATA) เมื่อมีสัญญาณข้อมูลเข้ามา ระดับสัญญาณ RS 485 จะถูกแปลงระดับสัญญาณลงให้อยู่ระดับสัญญาณแบบ TTL โดยใช้ IC # 75176 เป็นตัวแปลงระดับสัญญาณลง จากนั้นข้อมูลก็จะถูกนำเข้าไปเก็บใน SERIES BUFFER ของ MICROCONTROLLER แล้ว MICROCONTROLLER จะทำการตรวจข้อมูลชุดที่ส่งมาให้นั้นว่ามีตำแหน่ง (ADDRESS) ตรงกับ ANALOG OUTPUT UNIT หรือไม่ เมื่อตรวจสอบแล้วพบว่า ตำแหน่ง (ADDRESS) ตรงกับส่วนของ ANALOG OUTPUT UNIT ก็จะทำการรับข้อมูลนั้นเข้ามาแล้วทำการแปลงข้อมูลให้เป็นข้อมูลแบบขนานขนาด ๘ บิท ส่งออกไปทาง PORT 1 ต่อไป แต่ถ้าข้อมูลชุดที่ส่งมานั้นมีตำแหน่ง (ADDRESS) ไม่ตรงกับ ANALOG OUTPUT UNIT แล้ว MICROCONTROLLER จะไม่กระทำการใด ๆ กับข้อมูลชุดนี้อีก และที่ PORT 1 ค่าข้อมูลที่ปรากฏอยู่ก็จะยังคงเป็นค่าของข้อมูลเดิม ก่อนหน้าที่จะมีข้อมูลส่งมายังส่วน ANALOG OUTPUT UNIT แล้วถูก MICROCONTROLERปฏิเสธข้อมูลชุดนั้น ค่าที่ออกจาก Port 1 ของ MICROCONTROLLER ถูกส่งไปเข้าวงจร D TO A รับข้อมูลที่ส่งมาจากคอมพิวเตอร์ทาง PORT OUTPUT 8 Bit ของ MCS 51 เป็นสัญญาณดิจิตอลขนาด 8 BIT มาเข้าวงจร D TO A ซึ่งมีวงจรตามรูปวงจร D TO A ทำหน้าที่เปลี่ยนสัญญาณจากดิจิตอลเป็นอนาล็อค โดยเลือกใช้ IC DAC 0808 ซึ่งหาง่ายและราคาถูกตามรูป การทำงานโดย IC ตัวนี้มี INPUT ป้อนเข้าตั้งแต่ 5 - 12 ของ IC DAC 0808 OUTPUT ออกทางด้านขา 4 ส่วนขา 14 จะเป็นตัวปรับไฟ REFERANCE เพื่อควบคุมและรักษาระดับแรงดันของ OUTPUT ให้คงที่ และนำสัญญาณที่ขา OUTPUT ของ DAC 0808 ซึ่งเป็นสัญญาณกระแส โดยใช้ IC LF 351 เป็นแรงดัน โดยมี Rp1 และ Rp3 Rp1 ปรับ SPAN Rp3 ปรับ ZERO เพื่อควบคุมให้แรงดันออกมามีค่า 1 - 5 V จากนั้นก็นำแรงดัน 1 - 5 V มาเข้าวงจร V TO I เพราะทรานซิสเตอร์เบอร์ 2N 2222 ส่วนใหญ่ใช้ 4 - 20 mA ดังวงจรตามรูป โดยป้อนแรงดัน INPUT IC เข้ามา โดยใช้ IC LM 324 LIA ขา 2 เป็นวงจร NON INVERING แรงดันที่ออกมา ขา 4 เป็นลบและผ่านวงจร NON INVERING อีกตัว โดยใช้ IC LM324 L1B ขา 6 แล้วมีแรงดันออกมาที่ขา 7 เป็นบวกและจ่ายแรงดันไปที่ ขา 10 ของ IC LM234 L1C ซึ่งเป็น BUFFER ต่อให้ครบวงจร จากนั้นจะมี OUTPUT ออกมาที่ ขา 8 ต่อเข้ากับทรานซิสเตอร์ ซึ่งเป็นตัวขยายกระแสให้กับวงจร เพื่อที่จะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

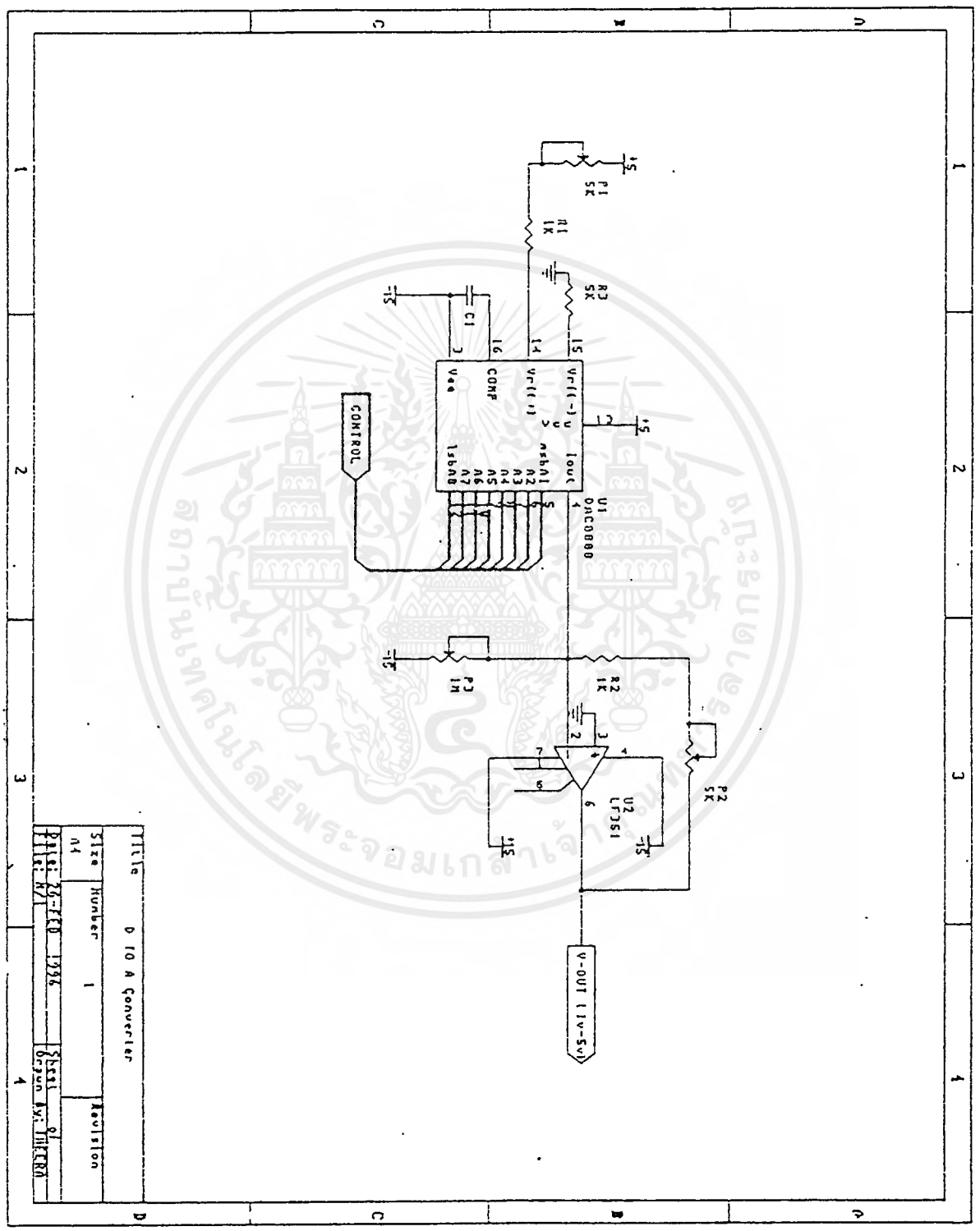
จ่ายกระแสให้ได้มาก ในวงจรนี้มี SPAN และ R ZERO เป็น R ปรับค่าได้ใช้ควบคุมแระแสในวงจร โดยกระแสที่ต้องการคือในช่วง 4 - 20 mA

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



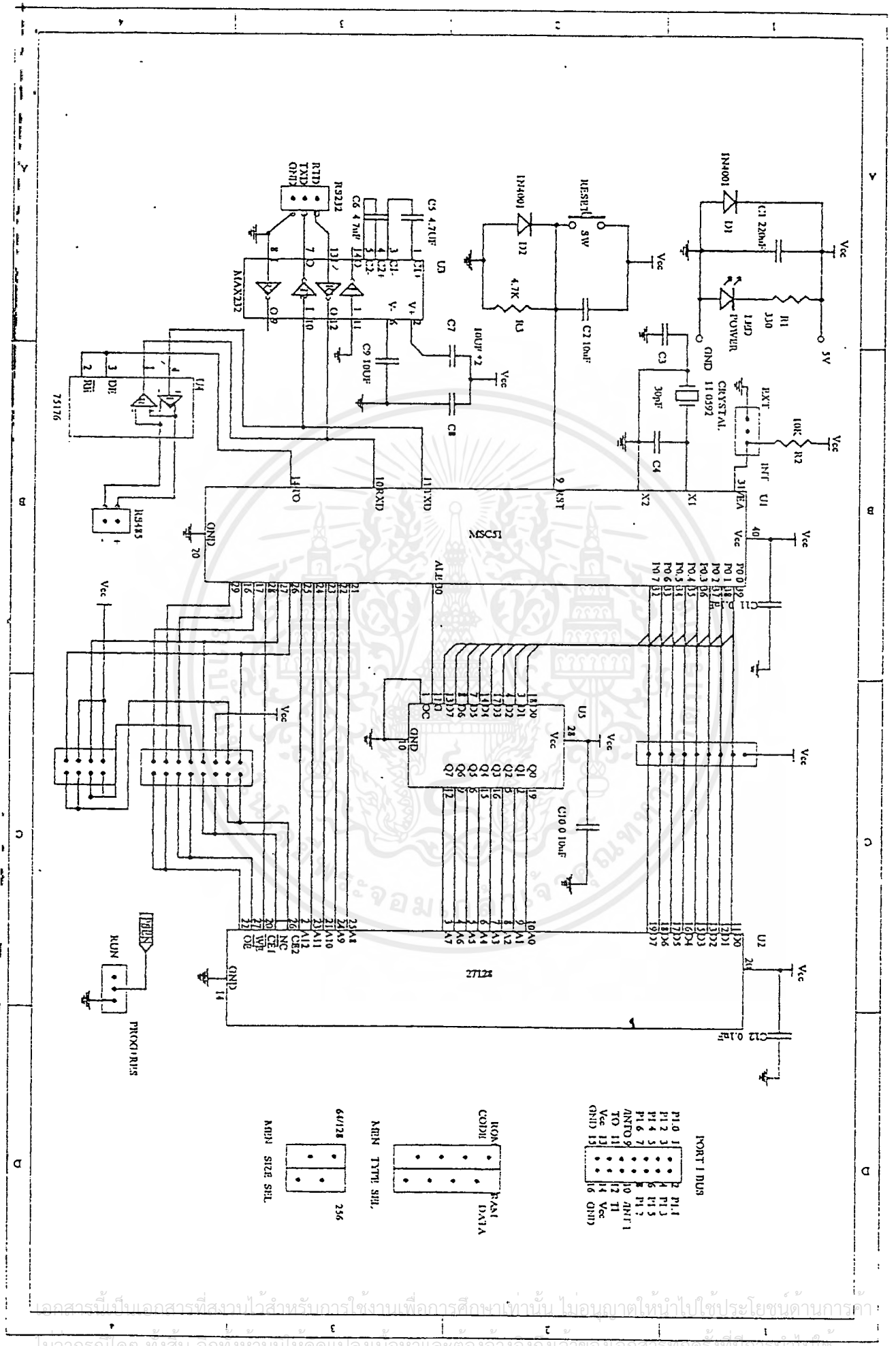
รูปที่แสดงวงจร VOLTAGE TO CURRENT

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่แสดงวงจร DIGITAL TO ANALOG

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทดลอง ANALOG OUTPUT

ชุดทดลอง

1. ชุด POWER SUPPLY ขนาด +5 V , -5 V , +15 V , - 15 V
2. ชุด CONTROLER ใช้ MCS 51 เป็นรับและแปลงค่า ให้ออกมาเป็น DIGITAL ขนาด 8 บิต
3. ชุดวงจร D/A ทำหน้าที่เปลี่ยน DIGIGAL ขนาด 8 บิตให้เป็น ANALOG
4. ชุดเปลี่ยนแปลงแรงดันเป็นกระแส 4 - 20 mA
5. ชุดทดลองการควบคุม LEVEL

ขั้นตอนการทดลอง

1. นำวงจรมาต่อรวมกัน แล้วนำสัญญาณควบคุมต่อเข้ากับ CONTROL VALVE และใช้มิ เตอร์จับไว้ที่ OUTPUT ANALOG
2. ต่อสายสัญญาณเข้ากับคอมพิวเตอร์ แล้วทำการส่งค่าเป้าหมายมาควบคุม CONTROL VALVE ที่ 0 %, 25 %,50 % ,75 % และ 100 %
3. ทำการเปลี่ยนค่าเป้าหมายไปเรื่อย ๆ และคอยปรับ R เพื่อหาค่าที่เหมาะสมที่จะทำให้เกิดค่าผิดพลาดน้อยที่สุด
4. จากการทดลองสามารถจ่ายโหลดได้สูงสุดประมาณ 1.2 กิโลโอห์ม โดยการทดลองให้ VR มาต่อแล้วปรับค่าไปเรื่อย ๆ จนเกิดความเพลี้ยนมากที่สุด แต่จะได้ ความเที่ยงตรงช่วง 0 - 750 โอห์ม
5. จากการทดลองมากมายหลาย ๆ ค่า แต่เราเฉลี่ยมาให้เห็นเพียงค่า ค่าที่วัดขึ้นและลง 1 รอบเท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางผลการทดลองของ อนาล็อกเอ้าท์พุท

เป้าหมาย (%)	OUT PUT D to A (VOLTAGE)	OUT PUT V to I (mA)	% การเปิดของ CONTROL VALAE
0 %	1.00	4.00	0.00 %
25 %	2.03	8.08	25.02 %
50 %	3.05	12.08	50.03 %
75 %	4.06	16.03	75.04 %
100 %	5.00	20.00	100.00 %

เป้าหมาย (%)	OUT PUT D to A (VOLTAGE)	OUT PUT V to I (mA)	% การปิดของ CONTROL VALAE
100 %	5.00	20.00	100 %
75 %	4.05	16.03	75 %
50 %	3.06	12.06	50 %
25 %	2.05	8.08	25 %
0 %	1.00	4.00	0 %

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปัญหาที่เกิดขึ้น

การไม่ค่อยเสถียรภาพของค่าต่าง ๆ ในวงจร เช่น แรงดันไฟฟ้าและกระแสไฟฟ้า ทำให้ต้องมีการปรับแต่งอยู่เสมอและบางครั้งก็เกิดการผิดพลาดบ้าง แต่ก็ยังคงอยู่ในช่วงที่สามารถยอมรับได้

ในส่วนของการควบคุมนั้น มีปัญหาในส่วนที่ว่าเราจะใช้ค่าของพารามิเตอร์ต่าง ๆ ในการควบคุมเป็นค่าเท่าใดในสภาวะปกติเพื่อให้การควบคุมอยู่ในสภาพที่ดีที่สุด ซึ่งทางผู้จัดทำอาศัยการทดลองแล้วเปลี่ยนไป เพื่อดูผล แต่การควบคุมจริงสามารถตั้งค่าได้

จะเห็นได้ว่าปัญหาส่วนใหญ่ที่เกิดขึ้นนั้น ส่วนใหญ่จะมาจากการติดต่อเชื่อมโยงการทำงานในแต่ละส่วนเข้าด้วยกัน ซึ่งเราต้องมีการเสียเวลาในการทดลอบและแก้ไขเพื่อให้การทำงานได้มีความถูกต้องมากขึ้น

เช่น ถ้าค่าที่ out put ของ IC 1408 เมื่อเราป้อน DATA 00 และ FFH ค่าต่างกันน้อยมาก จะทำให้ค่าที่จะออกมาเป็นสัญญาณ 1 - 5 โวลต์ เกิดการไม่ลีเนีย ขึ้นมาในช่วง CA - FA จะขึ้น ๆ ลง ๆ เราต้องควบคุมไฟเข้าที่ขา ref ให้ค่า 00 และ FF ของค่าให้มีช่วงที่เหมาะสมจึงจะได้ค่าตามที่ต้องการ

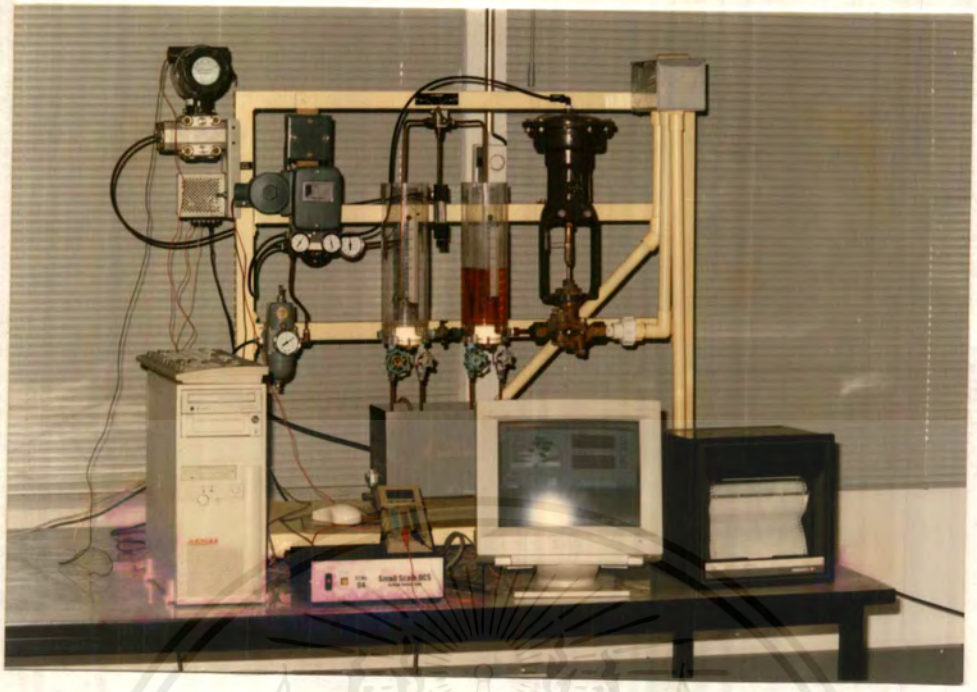
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สรุป

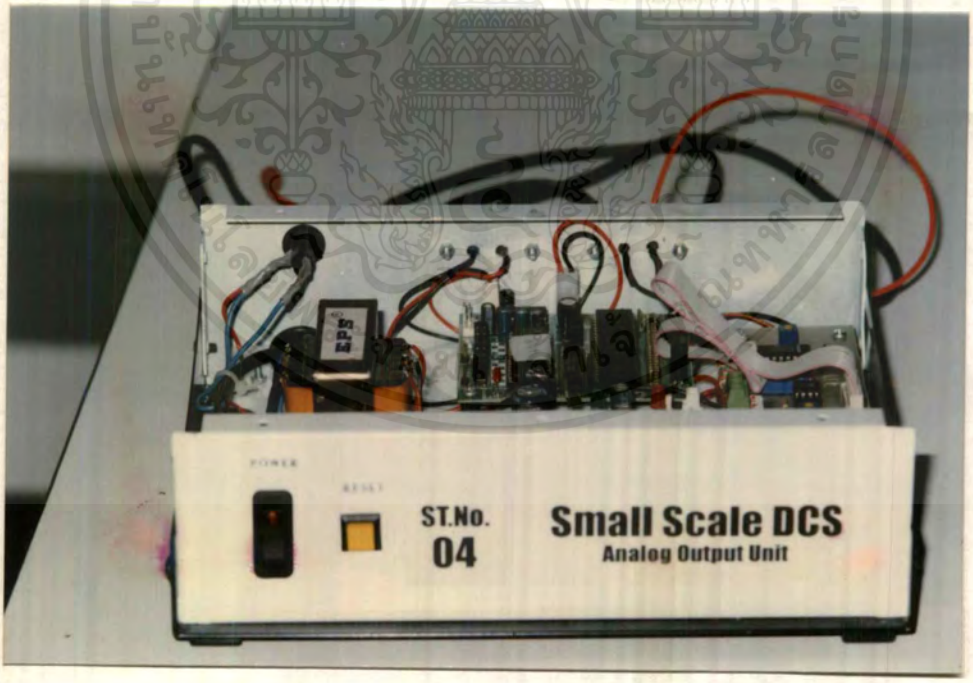
ส่วนของฮาร์ดแวร์ที่สร้างขึ้นนั้นประกอบด้วย วงจรแปลงสัญญาณดิจิตอล เป็นสัญญาณอนาล็อก วงจรแปลงกระแสไฟฟ้าเป็นแรงดันไฟฟ้า โดยในส่วนของวงจรแปลงสัญญาณดิจิตอลเป็นสัญญาณอนาล็อก ได้ทำการทดสอบเพื่อปรับค่าให้ได้ตามที่กำหนด คือ จะต้องได้เอาต์พุท 1 - 5 โวลต์ ทดลองโดยการใช้สัญญาณดิจิตอลจากแจท 31 จำลองสัญญาณดิจิตอลขึ้น ซึ่งจะมีค่าอยู่ระหว่าง 00 - FFH ผลจากการทดลอง วงจรแปลงสัญญาณดิจิตอลเป็นสัญญาณอนาล็อก จากการปรับค่า R ก็จะได้สัญญาณอนาล็อกสามารถทำงานได้ถูกต้อง โดยจะให้เอาต์พุทขนาด 0.99 - 4.99 Volt เมื่อสัญญาณดิจิตอลที่ 00 - FFH ซึ่งผลที่ได้มีค่าที่ใกล้เคียงมาก จากผลการทดลองได้ค่าดังนี้

ในส่วนของวงจรแปลงแรงดันไฟฟ้าเป็นกระแสไฟฟ้า เพราะว่าใน Transmitter ของโรงงานอุตสาหกรรม จะใช้รับสัญญาณควบคุมในช่วง 4 - 20 mA ดังนั้นเราจึงต้องทำการแปลงแรงดันไฟฟ้าเป็นกระแสไฟฟ้า ตามวงจรข้างล่างนี้ และปรับค่าให้ออกมา 4 - 20 mA เมื่อมีอินพุท 1 - 5 Volt ในส่วนวงจรแปลงแรงดันไฟฟ้าเป็นกระแสไฟฟ้านั้น ซึ่งจะแปลงแรงดัน ขนาด 1 - 5 โวลต์ ให้เป็นกระแสขนาด 4 - 20 mA เราทำการทดลองโดยต่อ เอาต์พุทของวงจรแปลงสัญญาณดิจิตอลเป็นสัญญาณอนาล็อก ขนาด 1 - 5 โวลต์ มาป้อนเป็นอินพุทแล้วทำการปรับ ผลที่ได้จากวงจรจะให้เอาต์พุทที่ 4 - 20 mA

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

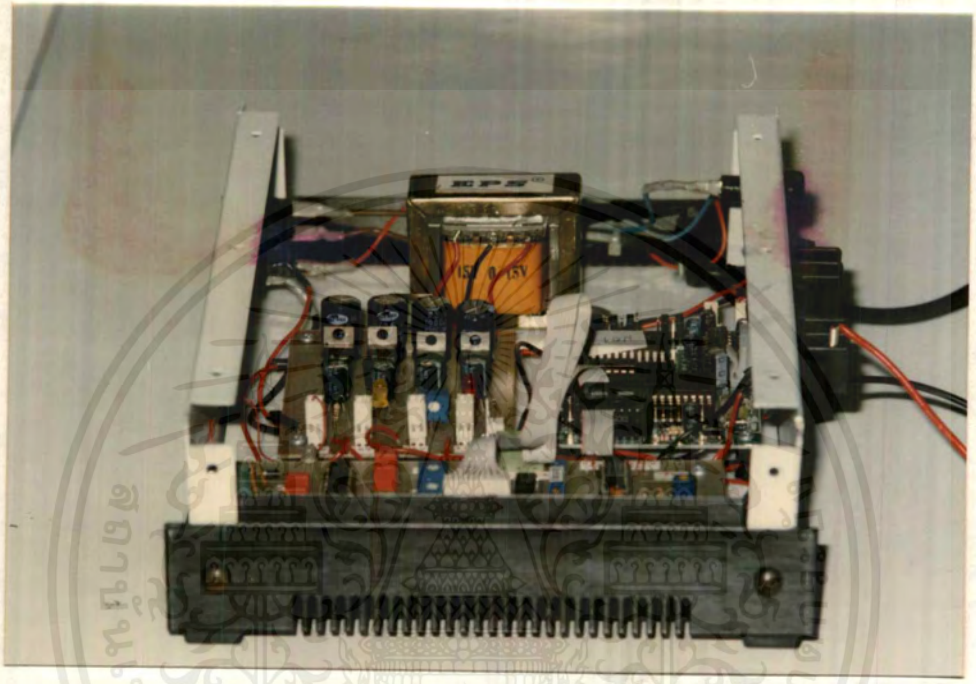


รูปแสดง UNIT 04 (ANALOG OUTPUT UNIT) ทำการต่อควบคุมกระบวนการระดับน้ำ



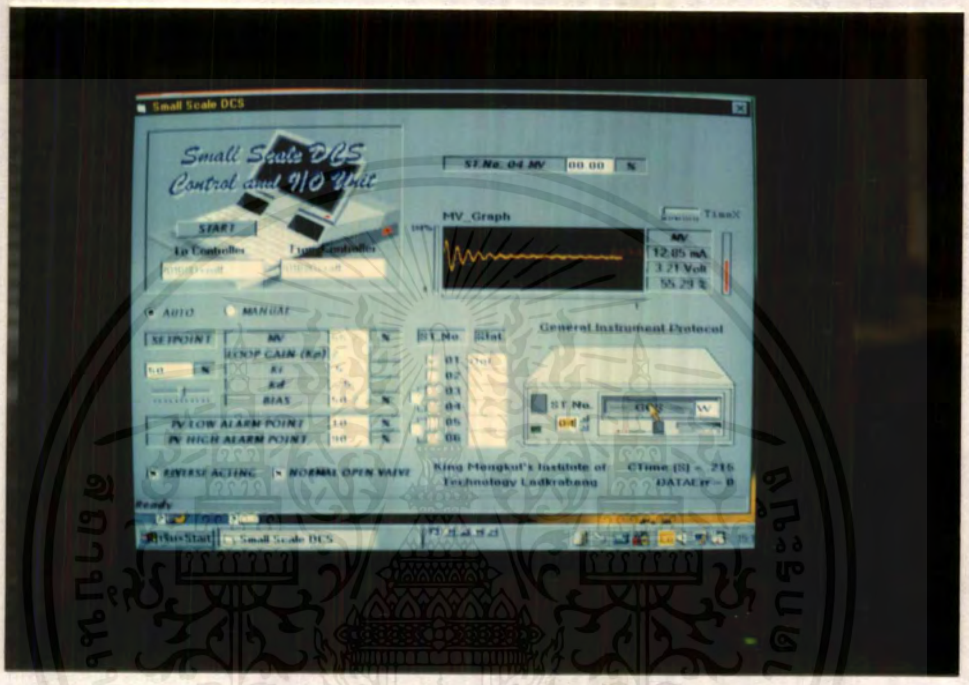
รูปแสดง UNIT 04 (ANALOG OUTPUT UNIT)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



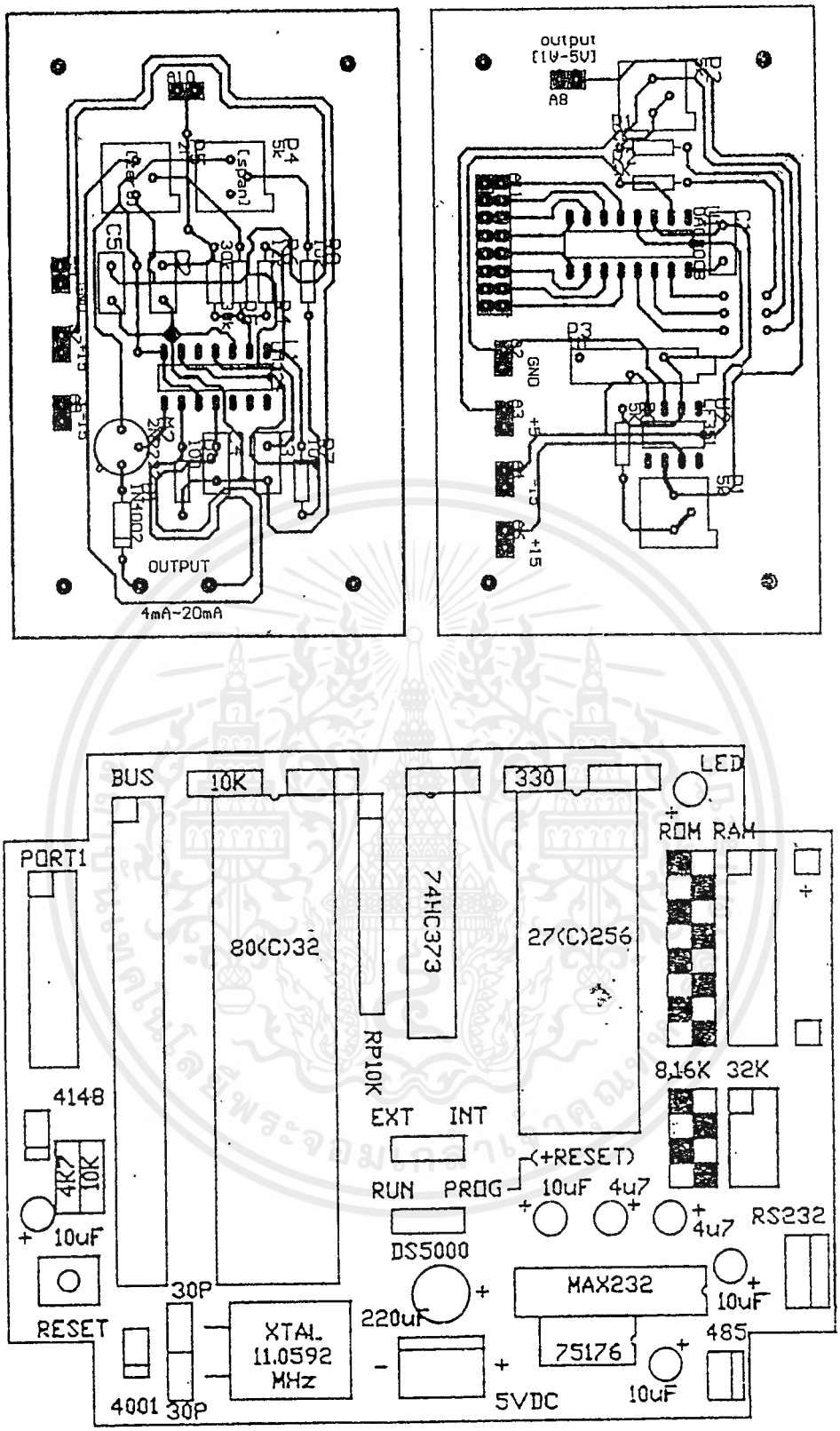
รูปแสดงการต่อวงจรภายในของ UNIT 04

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปแสดงหน้าจอของ ST.No. 04

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ แสดง การวางอุปกรณ์ DIGITAL TO ANALOG

VOLTAGE TO CURRENT

CONTORLER MCS 51

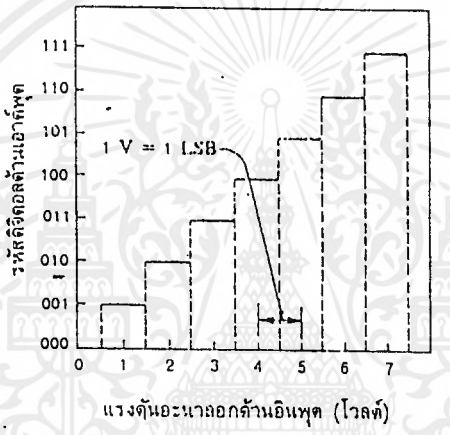
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ANALOG TO DIGITAL CONVERTOR

กระบวนการต่างๆที่เกิดขึ้นตามธรรมชาติส่วนใหญ่หากนำมาแปรค่าตามสัญญาทางไฟฟ้ามักเป็นสัญญาณที่อยู่ในรูปของแรงดันของกระแส หรือไม่ก็เป็นลักษณะของค่าความต้านทาน ลักษณะที่ได้จะเป็นสัญญาณอนาลอก ซึ่งไม่สามารถจะนำไปใช้กับคอมพิวเตอร์โดยตรงได้จึงจำเป็นต้องมีวงจรแปรสัญญาณอนาลอกเป็นสัญญาณดิจิตอล เราเรียกว่าวงจรที่ทำหน้าที่ดังกล่าวว่า Analog to Digital Convertor

หลักการเบื้องต้นของวงจร Analog to Digital Convertor

หากนำเอา Analog to Digital Convertor ขนาด 3 bit มาเขียนกราฟคุณสมบัติระหว่างสัญญาณ input กับ output สมมติว่าแรงดัน input Vi เปลี่ยนค่าจาก 0 - 7 Volt และได้สัญญาณ output ที่เป็นสัญญาณ Digital จาก 000 - 111 ดังแสดงในรูปที่ 1



รูปที่ 1 แสดงกราฟคุณสมบัติของ ADC 3 bit

ค่าความละเอียดของ Analog to Digital Convertor หาได้จากการเปลี่ยนแปลงค่าแรงดัน input แล้วทำให้สัญญาณ Digital เปลี่ยนค่า bit นัยสำคัญต่ำสุดไป

ความละเอียด = ค่าแรงดัน input ต่อ bit = ค่าเต็มสเกลหารด้วย $2^n - 1$

หรือถ้าอ้างถึงเรื่อง Digital to Analog ที่กล่าวมาแล้วจะได้ว่า

ความละเอียด = 2^n

ถ้า n คือจำนวน bit ของวงจร

จากรูปที่ 1 จะเห็นว่าขณะ output เป็น 001 แรงดัน input มีค่าเท่ากับ 1 Volt ซึ่งค่านี้เกิดจากแรงดันค่าเฉลี่ยของ 0.5 กับ 1.5 Volt หรืออาจกล่าวได้ว่าขณะ output เป็น 001 แรงดัน input ถูกกำหนดให้จะอยู่ในช่วง 0.5 - 1.5 Volt ซึ่งจะมีค่าผิดพลาดเท่ากับครึ่ง bit ดังนั้นหากต้องการให้ค่าผิดพลาดนี้ลดลงจำเป็นต้องเพิ่มจำนวน bit ให้สูงขึ้น

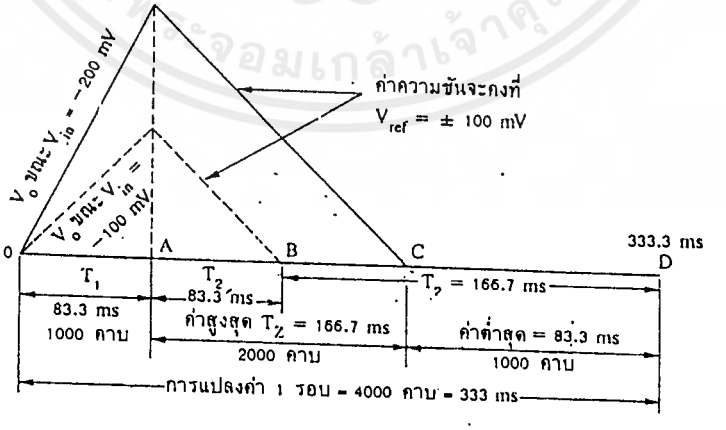
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วิธีการเปลี่ยนสัญญาณ Analog ให้เป็น Digital นั้นมีมากมายหลายแบบ หากแบ่งตามความเร็วที่ใช้ในการเปลี่ยนสัญญาณมี 3 แบบ ดังแสดงคุณสมบัติของแต่ละแบบในตารางที่ 1

แบบ	ความเร็ว	ช่วงเวลาแปลงสัญญาณใน 1 รอบ	การใช้งาน
รวบรวมค่า (intergrating)	ช้า	มิลลิวินาที	DC Voltmeter
ประมาณค่าต่อเนื่อง(successive approximation)	เร็ว	ไมโครวินาที	สัญญาณเสียง
แฟลช (flash)	เร็วมาก	นาโนวินาที	สัญญาณภาพ

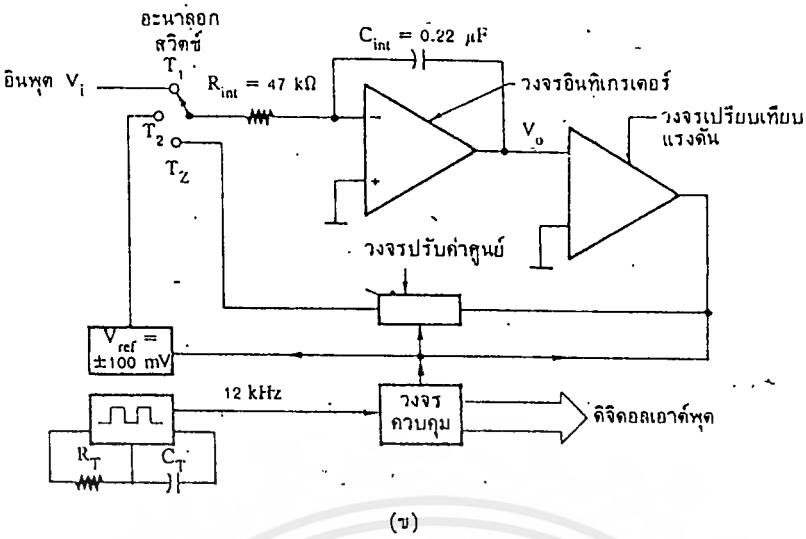
Analog to Digital ที่ใช้งานทั่วไปมักเป็น 2 แบบแรก เนื่องจากมีวงจรค่อนข้างง่าย ราคาถูกต่ก็มีความแม่นยำพอควร สามารถนำมาใช้งานทั่วไปได้

Integrating Analog to Digital Convertor(แบบรวบรวมค่า) แบบนี้เป็นแบบที่ใช้งานเกี่ยวกับเครื่องมือวัดความเร็วต่ำวงจรภายในประกอบด้วยวงจรที่เป็นแบบ Analog และ Digital รวมกันอยู่ใน IC ตัวเดียว เช่นเบอร์ 7106 และ 7107 เป็นต้น



(ก)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



(ข)

รูปที่ 2 แสดงแผนผังของวงจร ADC

การทำงานของวงจร

จากรูปที่ 2 สัญญาณ Clock ความถี่ 12 kHz ปรับความถี่ได้โดยการปรับค่า Rt และ Ct วงจรควบคุมจะควบคุมการโยกสวิตช์ และแปลงค่าแรงดันอินพุต Vi ให้เป็นสัญญาณ Digital แล้วส่งค่าที่ได้ไปยังเอาต์พุต โดยแบ่งการทำงานได้เป็น 3 ช่วงเวลาด้วยกันคือ T1 , T2 , Tz

ช่วงเวลา T1 เป็นเวลาที่วงจรรับสัญญาณมาจากอินพุตเข้าไปยังวงจรอินดิเกรเตอร์ ซึ่งเป็นเวลาเก็บรวบรวมค่าแรงดันอินพุต ช่วงเวลา T2 เป็นช่วงเวลาสำหรับเก็บรวบรวมค่าสัญญาณอ้างอิงเทียบกับสัญญาณที่ได้จากอินพุต ส่วนช่วงเวลา Tz นั้นเป็นช่วงเวลาที่วงจรต้องปรับค่าต่างๆ ให้เป็นค่าเริ่มต้นแบบอัตโนมัติ ซึ่งจะได้อธิบายเพิ่มเติมเพื่อให้เกิดความเข้าใจมากขึ้นดังต่อไปนี้

ช่วงเวลา T1 อนาลอกสวิตช์จะถูกควบคุมจากวงจรควบคุมให้โยกไปยังตำแหน่ง T1 เพื่อรับค่าจากภายนอกเข้ามาเก็บรวบรวมค่าจากวงจรอินทิเกรเตอร์ได้สัญญาณ Vo อาจมีศักย์เป็นบวกหรือลบก็ได้ขึ้นอยู่กับสัญญาณอินพุตที่ป้อนเข้ามา ถ้าอินพุตมีศักย์เป็นลบจะได้ Vo เป็นบวก ดังแสดงในรูปที่ 2(ก) ช่วงเวลา T1 จะถูกควบคุมด้วยสัญญาณ Clock จำนวน 1000 คาบ หากใช้ความถี่ Clock 12 kHz จะหาค่าเวลา 1 คาบ ได้เท่ากับ 83.33 ไมโครวินาที ดังนั้น T1 ต้องมีค่าเท่ากับ 83.33 * 10E-6 วินาทีคูณด้วย 1000 เท่ากับ 83.33 มิลลิวินาที ค่า Vo ที่ได้จะแปรค่าเป็นสัดส่วนกับ Vi เสมอ

ช่วงเวลา T2 วงจรควบคุมจะควบคุมอนาลอกสวิตช์ให้เปลี่ยนตำแหน่งไปที่ T2 เป็นช่วงเวลาที่วงจรอินทิเกรเตอร์ต่อกับวงจรอ้างอิงที่มีขนาดแรงดัน 100 มิลลิโวลต์ ค่านี้ได้จากการประจุ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คาพาซิเตอร์อ้างอิง (ไม่ได้แสดงไว้ในวงจร) ในช่วงเวลา T1 แรงดันนี้จะมีศักย์ตรงกันข้ามกับแรงดันอินพุตเสมอ เมื่อสัญญาณผ่านวงจรอินทิเกรเตอร์ จะแปรค่าเป็นอัตราส่วนคงทีเสมอโดยจะเพิ่มหรือลดค่าลงไปเรื่อยๆ จนกระทั่งค่า Vo เป็น 0 หากค่า Vi ที่ป้อนเข้ามาทางอินพุตในช่วงเวลา T1 มีศักย์เป็นลบ และได้ค่าเวลา T2 แปรค่าตาม Vo และ Vi หาได้จากสูตร

$$T_2 = T_1\left(\frac{V_i}{V_{ref}}\right)$$

T_1 มีค่า 83.33 มิลลิวินาที V_{ref} มีค่า 100 มิลลิโวลต์ จะได้ค่า

$$T_2 = \left(\frac{83.33ms}{100mV}\right)V_i$$

การแปลงสัญญาณอนาลอกเป็นสัญญาณดิจิตอลนั้นจะเกิดขึ้นในช่วงเวลา T2 วงจรนับจะเริ่มนับเมื่อเวลา T2 เริ่มขึ้น และหยุดนับเมื่อเวลา T2 หมดลง จำนวนคาบที่ได้จะเป็นเอาพุตหาได้จากสูตร

$$Nx = \frac{\text{จำนวนคาบที่นับได้ในช่วงเวลา T2} * \text{ช่วงเวลา T2}}{\text{เวลา 1 วินาที}}$$

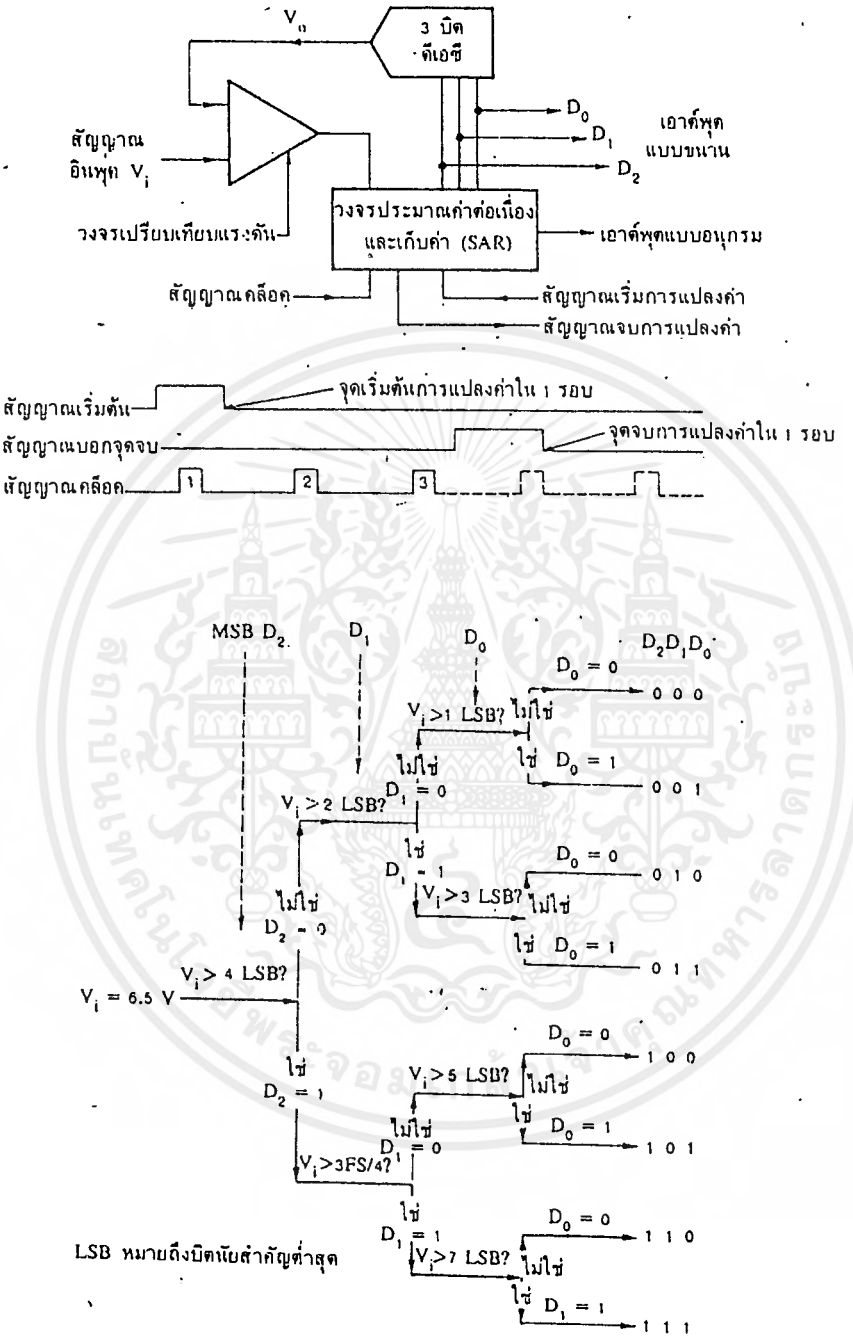
กำหนดให้ Nx คือ จำนวนคาบที่นับได้ในช่วงเวลา T2

ช่วงเวลา Tz เป็นช่วงเวลาที่ต้องปรับค่าเริ่มต้นให้กับวงจรภายในเช่น มีคาพาซิเตอร์ Cz ต่อขนานกับ Cint เพื่อแก้ข้อผิดพลาดของ Cint ในช่วงเวลา T1 และ T2 แรงดันใน Cz จะคอยช่วยปรับค่าแรงดันให้กับ Cint เพื่อให้ได้ค่าเริ่มต้นให้ถูกต้องในทุกกระบวนการแปลงค่าทำให้ผลลัพท์ที่ได้มามีค่าความแม่นยำสูงขึ้น จากรูปที่ 2 แสดงให้เห็นว่า ใน 1 รอบของการแปลงค่าประกอบด้วยสัญญาณ Clock จำนวน 4000 คาบ ใช้เวลา 333 มิลลิวินาที ช่วงเวลา T1 ใช้ Clock จำนวน 1000 คาบ ช่วงเวลา T2 และ Tz รวมกันไม่เกิน 3000 คาบ T2 แปรค่าตาม Vi ถ้าหากว่า Vi เป็น 0 จำนวน Clock ในช่วงเวลา T2 จะสูงสุดในขณะป้อน Vi เป็นค่าสูงสุด (200 มิลลิโวลต์) ได้จำนวน Clock ในช่วงเวลา T2 เท่ากับ 2000 คาบ จำนวน Clock ในช่วงเวลา Tz ลดลงเหลือ 1000 คาบ

Successive Approximation Analog To Digital Convertor (แบบประมาณค่าต่อเนื่อง) แบบนี้จะประกอบด้วยวงจรดีเอซี วงจรเปรียบเทียบแรงดัน และวงจรรีจีสเตอร์ เก็บค่าที่ได้หลังจากการประมาณค่าอินพุตที่รับเข้ามา (SAR = Successive Approximation Register) โดยวงจรรีจิสเตอร์มีขาอินพุท 1 ขา และขาควบคุมอีก 3 ขา คือขาแรกเป็นค่าสัญญาณบอกจุดเริ่มกระบวนการแปลงค่า ขาที่ 2 เป็นขาเอาท์พุทบอกจุดจบกระบวนการแปลงค่า ขาที่ 3 เป็นอิน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

พุทรับสัญญาณ Clock สำหรับควบคุมกระบวนการแปลงค่าในแต่ละรอบ ขาเอาท์พุตจะให้สัญญาณดิจิตอลมีทั้งแบบอนุกรมและแบบขนาน ดังรูปที่ 3



รูปที่ 3 แสดงแผนผังและรูปคลื่นของวงจร ADC แบบประมาณค่าต่อเนื่อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานของวงจร

การแปลงค่าในแต่ละรอบจะเริ่มขึ้นเมื่อวงจรได้รับสัญญาณเริ่มต้น วงจรรีจิสเตอร์ส่งค่าดิจิตอลที่ได้ประมาณค่าแล้วออกไปยังวงจรดีเอซี เพื่อทำการแปลงค่าเป็นสัญญาณอนาลอก Vo ส่งกลับมาเปรียบเทียบกับค่า Vi จากอินพุท ว่าค่าใดมากกว่ากัน เพื่อนำไปปรับค่าสัญญาณดิจิตอลแต่ละ Bit ให้กับรีจิสเตอร์ให้ถูกต้องตรงกับค่าที่ป้อนเข้ามาทางอินพุท การเปรียบเทียบเริ่มจากบิทสูงมาบิทต่ำเสมอ เมื่อครบทุกบิทวงจร SAR จะส่งสัญญาณสิ้นสุดกระบวนการออกไปที่ขาควบคุมและได้สัญญาณดิจิตอลเอาท์พุตที่สัมพันธ์กับค่า Vi ทางด้านอินพุต

หลักการประมาณค่าจะทำโดยวิธีการชั่งน้ำหนัก สมมติว่า ADC เป็นขนาด 3 บิทเทียบค่าตัวถ่วงน้ำหนัก เป็น 4 กก. 2 กก. และ 1 กก. ตามลำดับ สำหรับลอจิกจาก 111 ถึง 000 สมมติว่า Vi เป็น 6.5 โวลต์เทียบกับน้ำหนัก 6.5 กก. ซึ่งเป็นค่าที่ต้องการหา นำของที่ไม่ทราบค่านี้ขึ้นวางบนตาชั่งเอาตัวถ่วงขนาด 4 กก.มาถ่วงตาชั่งน้ำหนัก 6.5 กก.มีค่ามากกว่า 4 กก .จะต้องเซตบิทสูงสุดของ SAR เป็นลอจิก 1 โดยใช้สัญญาณ Clock ควบคุม 1 พัลส์

ลอจิกเอาท์พุทของวงจร SARจะเป็น 100 หลังจากนี้หากเพิ่มตัวถ่วงที่มีขนาด 2 กก.เข้าไปจะได้น้ำหนักตัวถ่วงรวมเป็น 6 กก. เปรียบเทียบกับ 6.5 กก.ปรากฏว่าน้ำหนักที่ไม่ทราบค่ามากกว่าอีก วงจร SAR จะเซตต่อมาเป็นลอจิก 1 โดยใช้สัญญาณ Clock พัลส์ที่ 2 ทำให้เอาท์พุทของวงจร SAR เป็น 110 ครั้งหลังสุดนั้นทดลองเพิ่มน้ำหนักตัวถ่วงอีก 1 กก. เปรียบเทียบกับ 6.5 กก. ปรากฏว่าน้ำหนักที่ไม่ทราบค่าน้อยกว่า 7 กก. วงจร SAR จะเซตบิทต่ำสุดเป็นลอจิก 0 ด้วย Clock พัลส์ที่3 ได้ลอจิกเอาท์พุทเป็นลอจิก 110 ซึ่งเป็นค่าที่สัมพันธ์กับสัญญาณอนาลอกทางด้านอินพุท เวลาที่ใช้ในการแปลงค่าจะแปรผันตามจำนวนบิทและความถี่ของสัญญาณ Clock หาได้จากสูตร

$$T_c = T(n+1)$$

กำหนดให้,

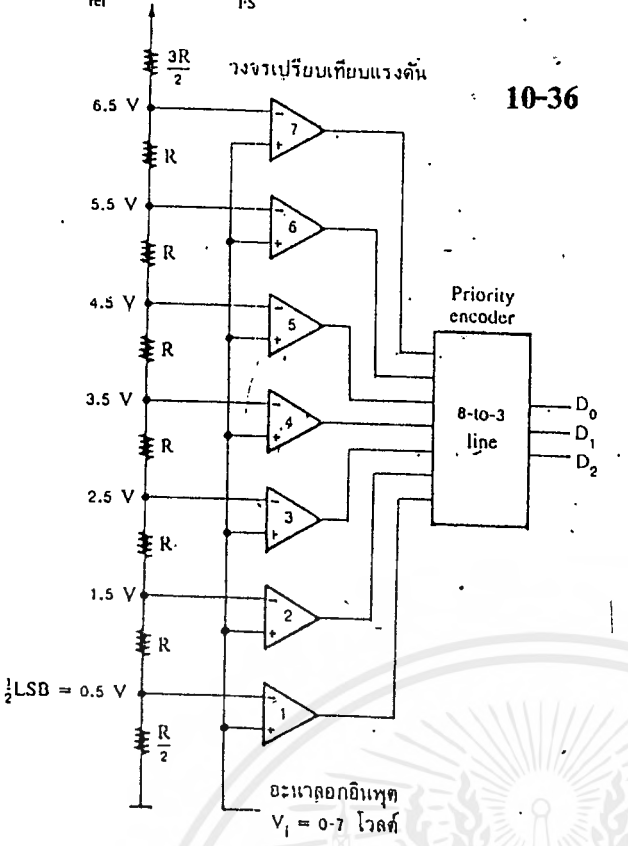
T_c คือค่าช่วงเวลาที่ใช้ในการแปลงค่าใน 1 รอบ

T คือค่าช่วงเวลา 1 คาบของสัญญาณ Clock

n คือค่าจำนวนบิทของวงจร

Flash Analog To Digital Convertor แบบนี้เป็นแบบที่มีความเร็วในการแปลงค่าสูงมาก ดังแสดงวงจรในรูปที่ 4 แรงดันอ้างอิงของวงจรเปรียบเทียบแรงดันจะถูกแบ่งค่าแรงดันด้วยตัวต้านทานเป็นขั้น ขั้นละ 1 โวลต์ เอาท์พุตของวงจรเปรียบเทียบแรงดันจะถูกต่อเป็นอินพุทของวงจรเอนโคดเดอร์แบบจัดลำดับ 8 ออก 3 ซึ่งลอจิกที่แปลงได้ทางเอาท์พุทจะแปลค่าตามอนาลอกทางด้านอินพุท ดังแสดงในตารางที่ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



อินพุต (v)	เอาท์พุต		
	D_2	D_1	D_0
0-0.5	0	0	0
0.5-1.5	0	0	1
1.5-2.5	0	1	0
2.5-3.5	0	1	1
3.5-4.5	1	0	0
4.5-5.5	1	0	1
5.5-6.5	1	1	0
>6.5	1	1	1

รูปที่ 4 แสดงวงจร ADC แบบแฟลซ และ ตารางที่ 2 แสดงความสัมพันธ์ อินพุทและเอาท์พุต

ช่วงเวลาสำหรับการแปลงค่าจะเร็วมากน้อยเท่าไรขึ้นอยู่กับผลตอบสนองของวงจรเปรียบเทียบแรงดันและวงจรเอนโคดเดอร์แต่ความเร็วจะสูงกว่า 2 แบบแรกที่กล่าวมาแล้ว สามารถนำมาใช้กับสัญญาณภาพหรือสัญญาณเรดาร์ได้ จากรูปที่ 4 จะเห็นว่าจำนวนของวงจรเปรียบเทียบแรงดันเป็น 7 ตัว ถ้าหากต้องการเอาท์พุท 3 บิท หาได้จากสูตร

$$\text{จำนวนวงจรเปรียบเทียบแรงดัน} = 2^n - 1$$

เมื่อ n คือจำนวนบิท

ดังนั้นถ้าเอาท์พุตเท่ากับ 3 บิท จำนวนวงจรเปรียบเทียบแรงดันจะมีทั้งหมด 7 วงจร

การตอบสนองต่อความถี่ทางสัญญาณอินพุทที่แปรค่าไปไม่เกิน $\pm\frac{1}{2}$ บิท ADC มีการตอบสนองต่อความถี่ของสัญญาณอินพุท Vi ที่มีการแปลค่าเป็นคลื่นรูปไซน์ช่วงไม่เกิน $\pm\frac{1}{2}$ บิท หาความถี่ได้ดังนี้

$$f_{max} = \frac{1}{2\pi\left(T_c\right)2^n}$$

กำหนดให้

f_{max} คือค่าความถี่สูงสุดที่ยอมได้ (Hz)

Tc คือช่วงเวลาในการแปลงสัญญาณ 1 รอบ

n คือจำนวนบิทของ ADC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

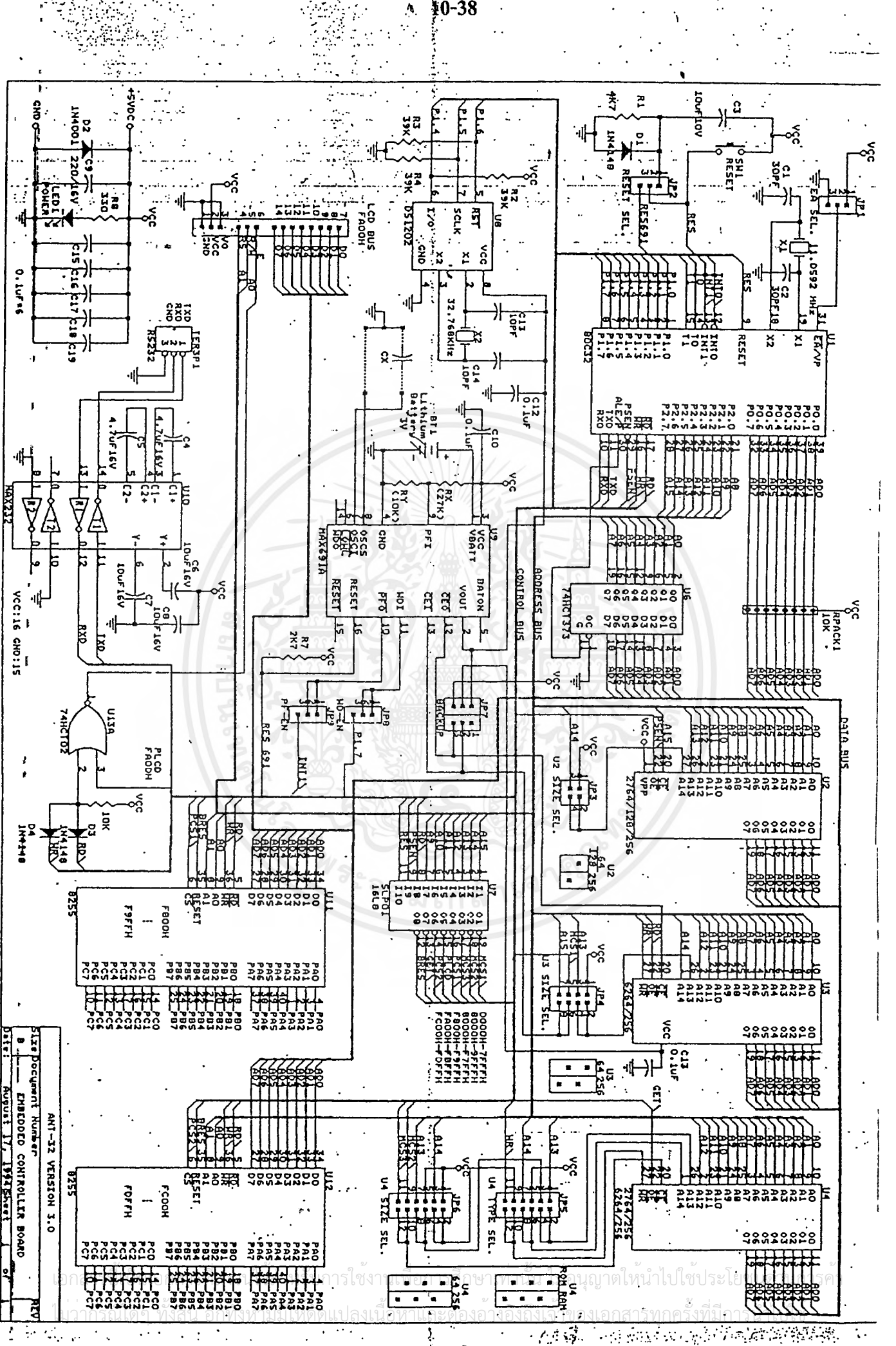
บทสรุป

จากบทความที่กล่าวมาทั้งหมด ได้แสดงถึงกระบวนการต่างๆ ที่เกิดขึ้นของตัวแปลงสัญญาณ A/D ตั้งแต่การทำงาน การส่งข้อมูลการกำหนดค่าสถานะต่างๆของขา ความแน่นอนและค่าความละเอียดที่เกิดขึ้น ฯลฯ ดังนั้นเราจึงสรุปหลักการต่างๆ ที่อธิบายมาทั้งหมดแยกเป็นหัวข้อต่างๆดังนี้

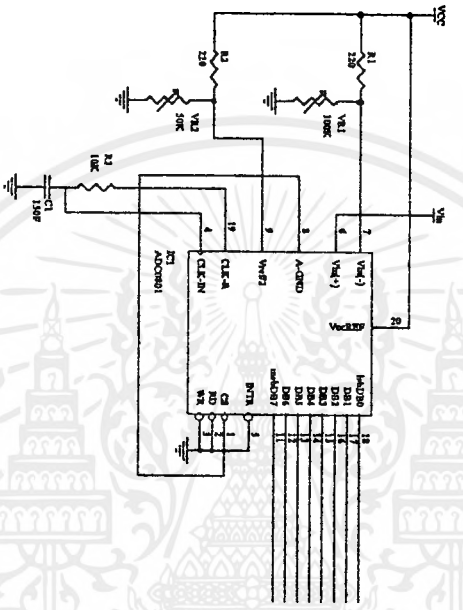
1. ตัวแปลงสัญญาณ A/D จะแปลงสัญญาณอนาลอกอินพุท เป็นสัญญาณดิจิตอลเอาท์พุท

2. ตัวแปลงสัญญาณ A/D จะใช้งานอยู่ที่อินพุทตั้งแต่ 0 - 5 โวลท์

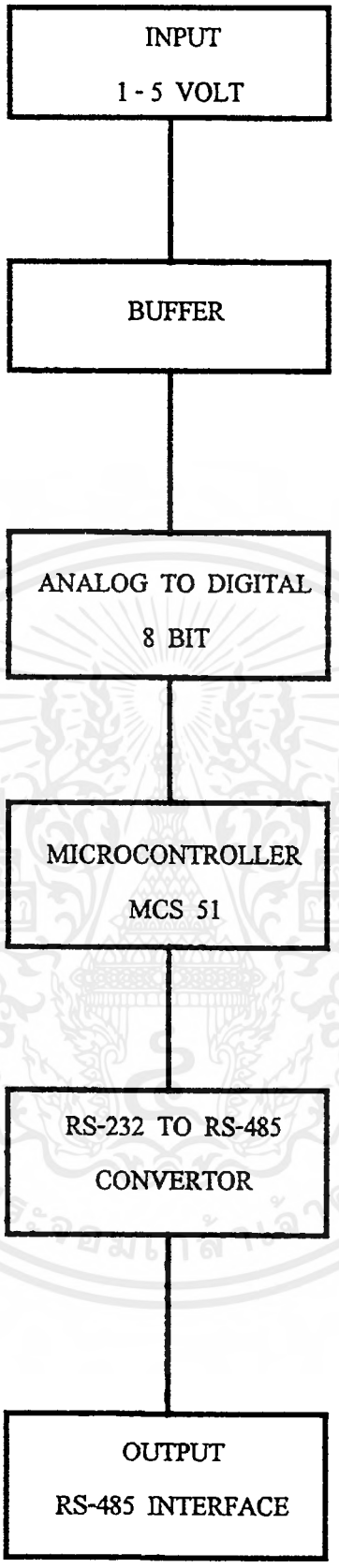
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่อนุญาตให้ทำซ้ำ ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของลิขสิทธิ์ของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



BLOCK DIAGRAM ANALOG INPUT UNIT

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้าที่ บล็อกต่างๆ

บล็อกที่ 1 มีหน้าที่ เป็นอินพุท 1 - 5 VOLT ซึ่งเป็นแรงดันที่ใช้ในโรงงานอุตสาหกรรมทั่วไป และขบวนการควบคุมต่างๆ

บล็อกที่ 2 มีหน้าที่ เป็นตัวกันแรงดันขาเข้าในกรณีที่ไม่คงที่

บล็อกที่ 3 มีหน้าที่ เปลี่ยนสัญญาณอนาลอกที่แรงดัน 1 - 5 โวลต์ เป็นสัญญาณดิจิตอล 8 บิทโดยใช้ IC ADC 0801

บล็อกที่ 4 มีหน้าที่ เป็น Main Board ซึ่งจะรับข้อมูลทางดิจิตอลขนาด 8 Bit ทาง Internal bus แล้ว CPU จะทำการแปลงค่าข้อมูลขนาด 8 Bit ที่รับเข้ามาเป็นข้อมูลอนุกรมขนาด 1 Bit ส่งออกทาง RS-232 แบบทีละ Bit

บล็อกที่ 5 มีหน้าที่ เป็นการแปลงสัญญาณ RS 232 เป็น สัญญาณ RS 485

บล็อกที่ 6 มีหน้าที่ เป็น OUTPUT RS 485

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้าที่หลักของ Analog Input Unit

ภาค ANALOG INPUT UNIT เป็นส่วนหนึ่งในหลาย ๆ ส่วนของระบบ SMALL SCALE DCS โดยในส่วนของ ANALOG INPUT UNIT มีลักษณะการทำงานดังนี้

ANALOG INPUT UNIT การทำงานเริ่มจากการป้อนแรงดันอินพุท 1 - 5 โวลต์ ซึ่งแรงดันอินพุทดังกล่าวจะถูกแปลงเป็นสัญญาณดิจิตอลโดย IC 0801 ซึ่งเป็น IC แปลงสัญญาณเป็นดิจิตอลขนาด 8 Bit เสร็จแล้วข้อมูลขนาด 8 Bit จะถูกส่งไปยัง Internal bus ของ Microcontroller หลังจากนั้น CPU จะทำการอ่านคำสั่งจาก EPROM ทำการแปลงข้อมูลขนาด 8 Bit ออกเป็นข้อมูล 1 Bit เพื่อส่งออกทาง PORT อนุกรม RS 232 แบบทีละ Bit

ข้อมูลทีละBit ทาง RS 232 จะถูกแปลงโดยใช้ IC 75176 เป็นตัวแปลงสัญญารทางด้าน RS 232 เป็นสัญญาณ RS 485 ซึ่งเป็นเอาท์พุทของ UNIT นี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทดลอง ANALOG INPUT

ชุดทดลอง

1. ชุด POWER SUPPLY ขนาด 1-5 V
2. ชุด CONTROLER ใช้ MCS 51 เป็นรับและแปลงค่า ให้ออกมาเป็น Digital 8 Bit
3. ชุดวงจร A/D ทำหน้าที่เปลี่ยน ANALOG ขนาด 1- 5 V เป็น Digital 8 Bit
4. ชุดวงจร RS 485 INTERFACE (232 TO 485)

ขั้นตอนการทดลอง

1. ต่อวงจร A/D ตามวงจร พร้อมทั้งวัดค่า OUTPUT DIGITAL ของวงจร A/D จะได้ผลดังตารางข้างล้างนี้

2. นำวงจรมาต่อรวมกัน แล้วโดยการนำเอาส่วน OUTPUT ของ A/D 8 Bit ต่อเข้า INTERNAL BUS ของชุด CONTROLER

3. จาก OUTPUT ของชุด CONTROLER การ INTERFACE จะเป็น RS 232 ทำการแปลงเป็น RS 485 ตามวงจรดังรูป

4. นำชุด A/D , ชุด CONTROLER และส่วนของ RS 485 INTERFACE มาต่อรวมกันแล้วทำการป้อน INPUT ที่ค่าต่างๆดังนี้คือ 1.5 , 2.45 , 3 , 4.5 , 5 V ตามลำดับและบันทึกผลในตาราง โดยดูผล OUTPUT 485 INTERFACE จอ MONITER ของคอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางผลการทดลองของ ANALOG INPUT

แรงดัน INPUT (Vdc)	OUTPUT Digital (HEX)
1	33
2	66
3	99
4	CC
5	FF

ตารางบันทึกผลการทดลองของวงจร Analog To Digital โดย IC 0801

แรงดัน INPUT (Vdc)	OUTPUT จาก COMPUTER
1.5	1.48
2.45	2.44
3	3.02
4.5	4.51
5	4.998

ตารางบันทึกผลการ ANALOG INPUT

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปัญหาที่เกิดขึ้น

การไม่ค่อยเสถียรภาพของค่าต่างๆ ในวงจร เช่น แรงดันไฟฟ้าและกระแสไฟฟ้า ทำให้ต้องมีการปรับแต่งอยู่เสมอ และบางครั้งก็เกิดการผิดพลาดบ้าง แต่ก็ยังคงอยู่ในสภาพที่ยอมรับได้

ในส่วนของการควบคุมนั้น มีปัญหาในส่วนที่เราจะใช้ค่าพารามิเตอร์ต่างๆ ในการควบคุมเป็นค่าเท่าใดในสภาวะปกติเพื่อให้การควบคุมอยู่ในสภาพที่ดีที่สุด ซึ่งทางผู้จัดทำอาศัยการทดลองแล้วเปลี่ยนไป เพื่อดูผล แต่การควบคุมสามารถตั้งค่าได้ เช่น ในส่วนของวงจร A/D ตามคุณสมบัติของ IC 0801 นั้นลักษณะของ INPUT ที่เข้ามามีค่าอยู่ในช่วง 0-5 Volt ทำการเปลี่ยนค่าเป็น 1-5 Volt โดยการให้แรงดัน 1 และ 2 Volt เข้าที่ขา 7 และ 9 ของ IC 0801 ซึ่งเมื่อนำมาประกอบรวมกันทั้งหมดต้องมีการปรับแต่งให้อยู่ในระดับ 1-5 Volt ซึ่งความไม่แน่นอนของค่าความต้านทานปรับค่าจะทำให้เกิดความยุ่งยาก

จะเห็นได้ว่าปัญหาส่วนใหญ่ที่เกิดขึ้นนั้น ส่วนใหญ่จะมาจากติดตั้งเชื่อมโยงการทำงานในแต่ละส่วนเข้าด้วยกัน ซึ่งเราต้องมีการเสียเวลาในการทดสอบและแก้ไขเพื่อให้การทำงานได้มีความถูกต้องมากขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สรุป

ส่วนของฮาร์ดแวร์ที่สร้างนั้นประกอบด้วย วงจรแปลงสัญญาณอนาลอกเป็นดิจิตอล ซึ่งมีการทำการเปลี่ยนแปลงวงจรเล็กน้อยคือ จาก 0-5 V เป็น 1-5 V ซึ่งเมื่อทำการแปลงวงจรแล้วได้อินพุตเป็น 1.02 - 4.998 V ซึ่งได้จากการปรับค่า R

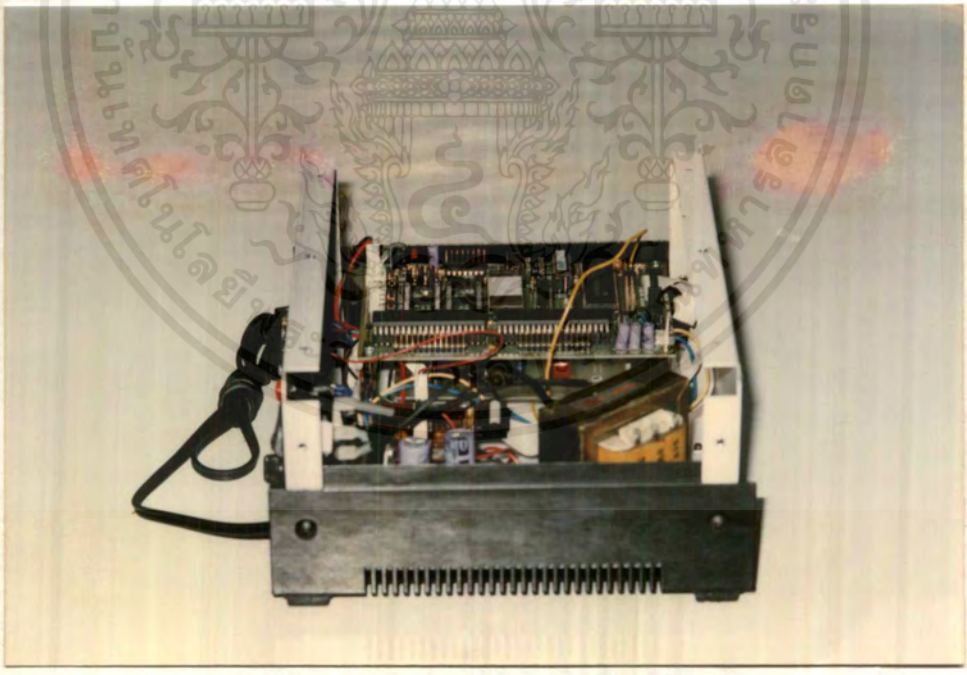
ในส่วนของ RS 485 INTREFACE ใช้ MAX 232 และ 75176 ซึ่งเป็น IC 232 INTERFACE และ 485 INTERFACE ตามลำดับ เนื่องจากเป็น IC สำเร็จรูปจึงไม่มีปัญหาในส่วนนี้

ในส่วนของไมโครคอนโทรเลอร์ ทางผู้จัดได้กำหนดให้ MCS 51 เป็นตัวลำเรียงข้อมูลโดยการอ่านคำสั่งจาก EPROM ซึ่งได้ทำการเขียนโปรแกรมเข้าไปเก็บไว้ในตัว EPROM แล้ว

จากทั้ง 3 ส่วนนำมาต่อรวมกันจะได้ดังข้อมูลตามตาราง

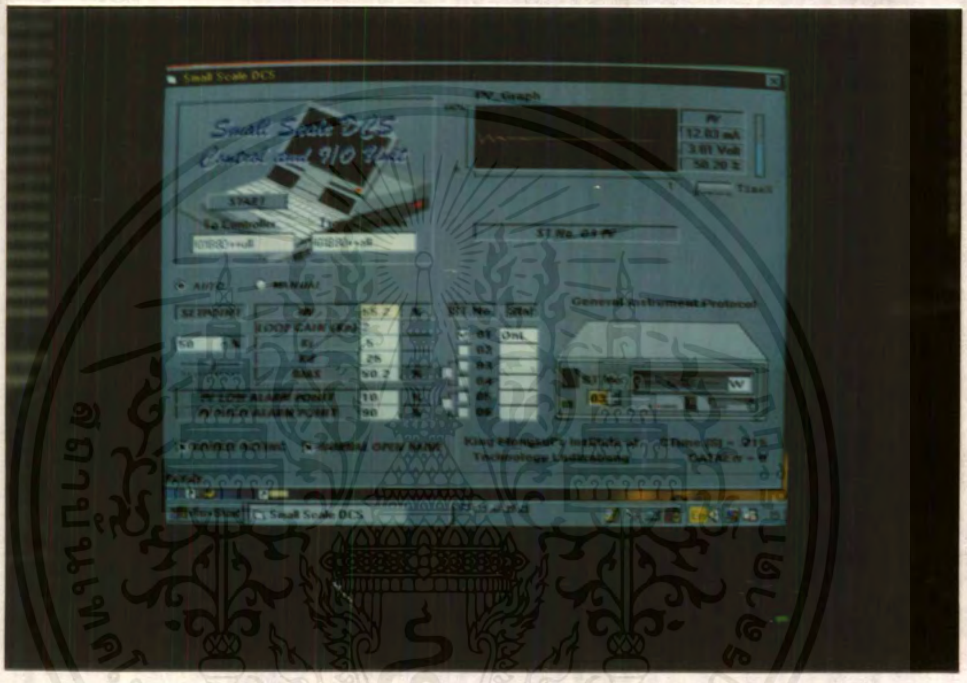
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปแสดง UNIT 03 (ANALOG INPUT UNIT)



รูปแสดงการต่อวงจรภายในของ UNIT 03

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปแสดงหน้าจอของ ST.No. 03

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

1.YOKOGAWA " DISTRIBUTED CONTROL SYSTEM "

2.สุเจตน์ จันทรังษ์ " SINGLE CHIP MICROCONTROLLER 8051 "

3.รศ.พิพัฒน์ เลาหสงคราม "ไมโครคอนโทรลเลอร์ MCS-48 MCS-51 "

4.KATSUHIKO OGATA " MODERN CONTROL ENGINEERING " , PRENTICE-HALL INTERNATION EDITIONS

5.CARLOS A.SMITH , ARMANDO B.CORRIPIO " PRINCIPLES AND PRACTICE OF AUTOMATIC PROCESS CONTROL " . JOHN WILLY & SONS

6J.MICHAEL JACOB " INDUSTRIAL CONTROL ELECTRONICS APPLICATIONS & DESIGN

7.JOE CAMPBELL " THE RS-232 SOLUTION "

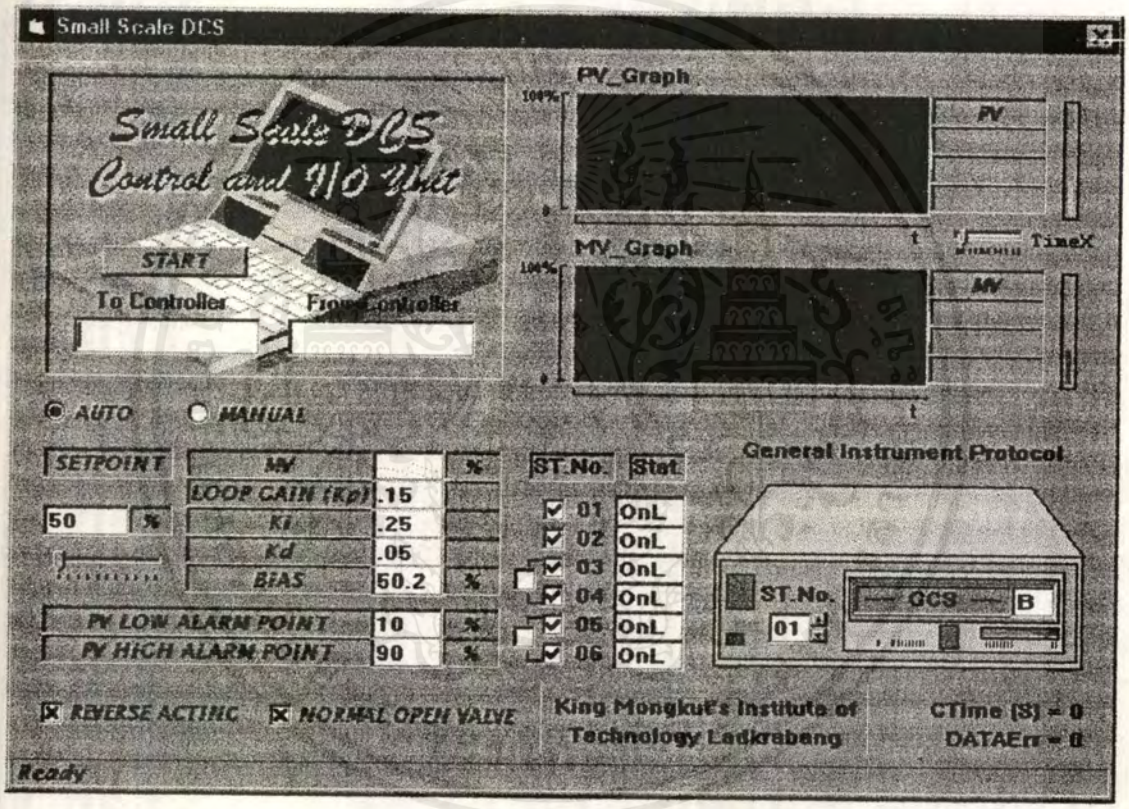
8PETER W.GOFTON " MASTERING SERIAL COMMUNICATIONS "

9.โอภาส ศิริครรชิตถาวร " OPTOCOUPLER DEVICES "

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

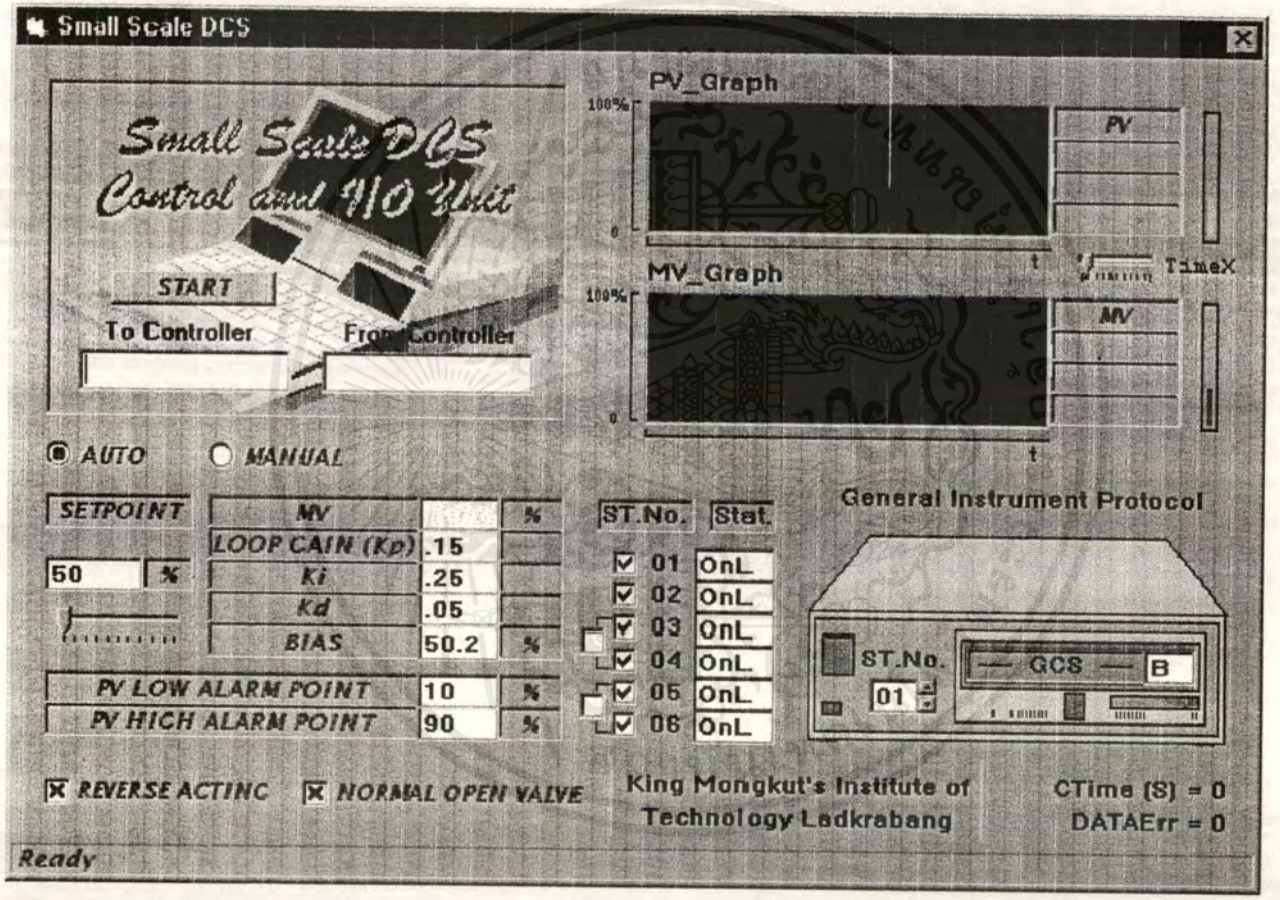
ภาคผนวก

ซอฟต์แวร์ลิสต์ของโปรแกรมที่ใช้ในระบบ



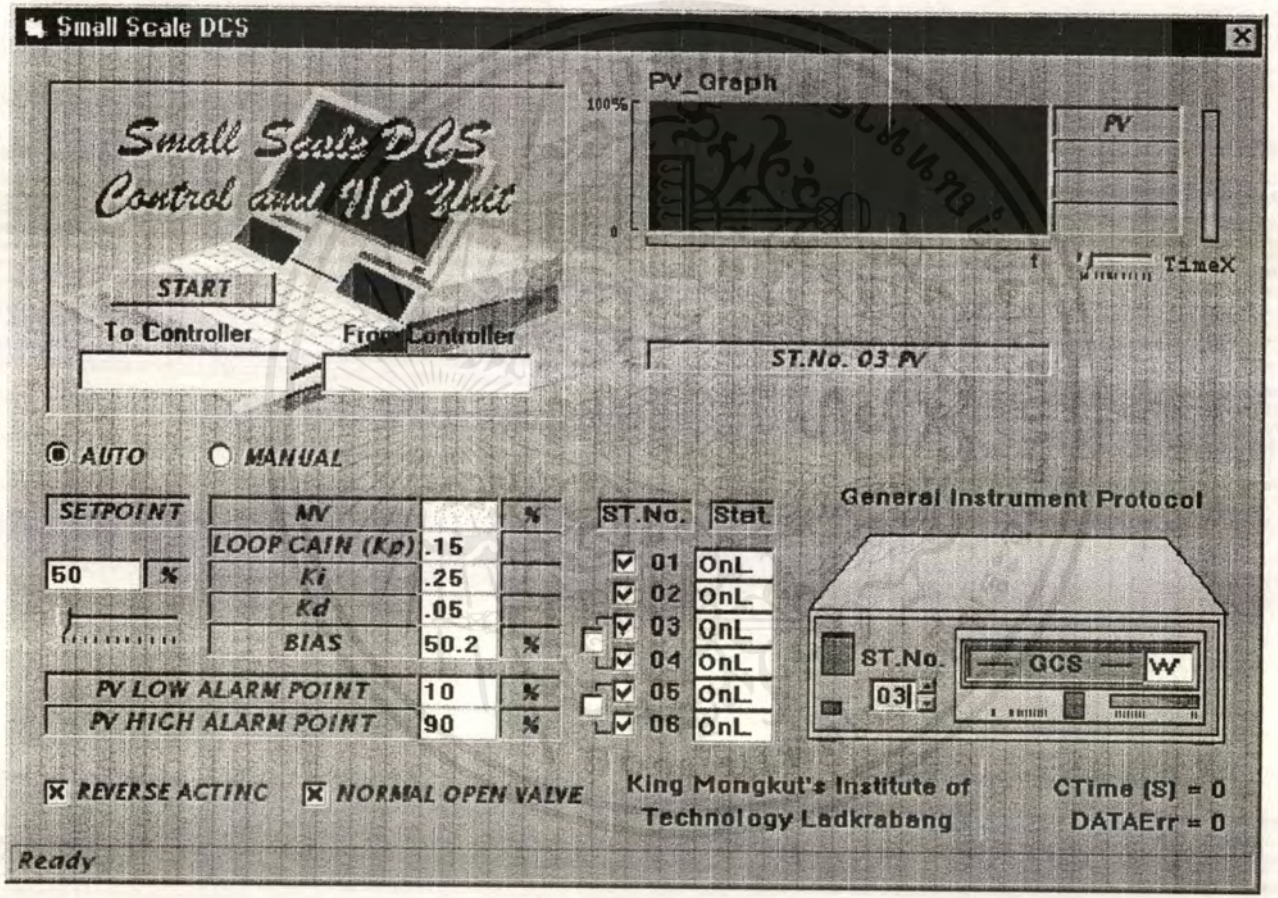
ภาพแสดงโปรแกรมมอนิเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



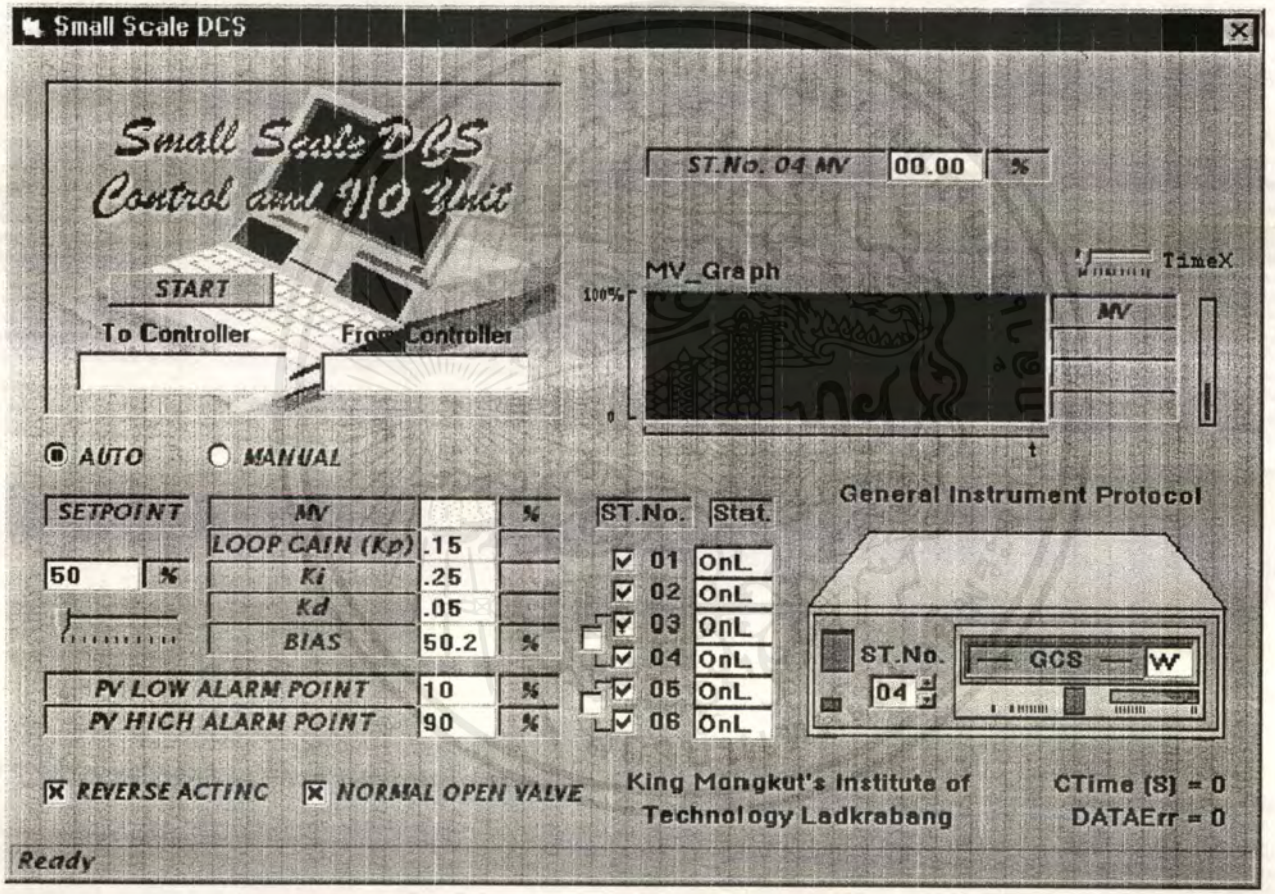
รูปที่ 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



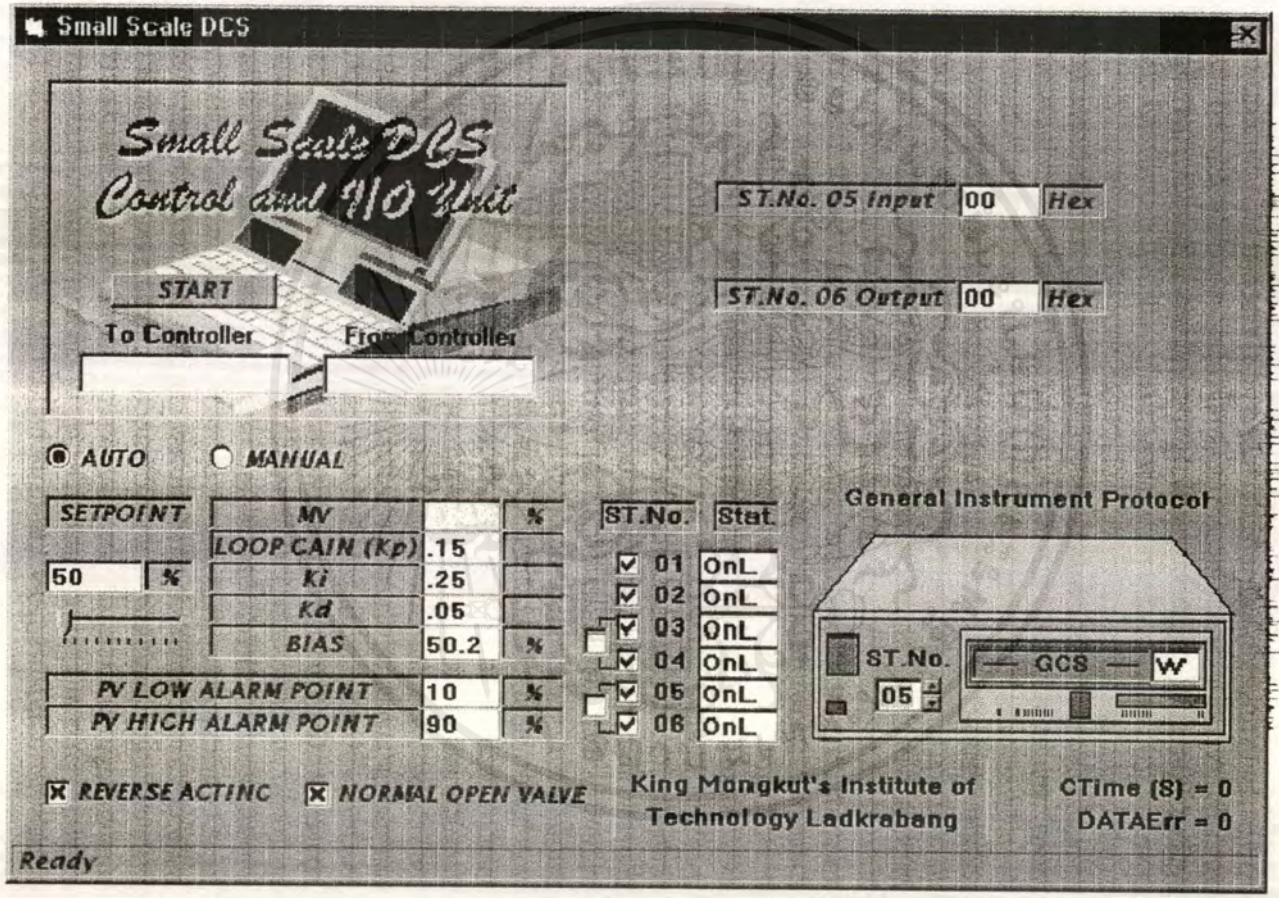
รูปที่ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


Small Scale DCS
Small Scale DCS
Control and I/O Unit
START
To Controller
From Controller
ST.No. 05 Input
00
Hex
ST.No. 06 Output
00
Hex
AUTO
MANUAL
SETPOINT
50
%
MV
%
LOOP CAIN (Kp)
.15
Ki
.25
Kd
.05
BIAS
50.2
%
PV LOW ALARM POINT
10
%
PV HIGH ALARM POINT
90
%
ST.No.
Stat.
01 OnL
02 OnL
03 OnL
04 OnL
05 OnL
06 OnL
General Instrument Protocol
ST.No.
05
GCS
W
REVERSE ACTING
NORMAL OPEN VALVE
King Mongkut's Institute of
Technology Ladkrabang
CTime (S) = 0
DATAErr = 0
Ready

รูปที่ 4

System Software CODE

Visual Basic 4.00 (32bit)

```
VERSION 4.00
Begin VB.Form DCSForm1
  AutoRedraw    = -1  'True
  BorderStyle   = 1  'Fixed Single
  Caption       = "Small Scale DCS"
  ClientHeight  = 6210
  ClientLeft    = 1560
  ClientTop     = 4275
  ClientWidth   = 9240
  FillColor     = &H0000FF00&
  FillStyle     = 0  'Solid
  BeginProperty Font
    name        = "Lucida Handwriting"
    charset     = 0
    weight      = 700
    size        = 8.25
    underline   = 0  'False
    italic      = -1  'True
    strikethrough = 0  'False
  EndProperty
  ForeColor     = &H00000000&
  Height        = 6615
  Left          = 1500
  LinkTopic     = "Form1"
  LockControls  = -1  'True
  MaxButton     = 0  'False
  MinButton     = 0  'False
  ScaleHeight   = 6210
  ScaleWidth    = 9240
  Top           = 3930
  Width         = 9360
  Begin VB.CheckBox Check8
    Caption     = "Check7"
    Height      = 225
    Left        = 4200
    TabIndex    = 86
    Top         = 4800
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    Width      = 255
  End
  Begin VB.CheckBox Check7
    Caption    = "Check7"
    Height     = 225
    Left       = 4200
    TabIndex   = 85
    Top        = 4320
    Width      = 255
  End
  Begin VB.CheckBox Check6
    Caption    = "06"
    BeginProperty Font
      name     = "MS SystemEx"
      charset  = 0
      weight   = 700
      size     = 7.5
      underline = 0  'False
      italic   = 0  'False
      strikethrough = 0  'False
    EndProperty
    Height     = 255
    Left       = 4440
    TabIndex   = 79
    Top        = 4920
    Value      = 1  'Checked
    Width      = 615
  End
  Begin VB.CheckBox Check5
    Caption    = "05"
    BeginProperty Font
      name     = "MS SystemEx"
      charset  = 0
      weight   = 700
      size     = 7.5
      underline = 0  'False
      italic   = 0  'False
      strikethrough = 0  'False
    EndProperty
    Height     = 255
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    Left        = 4440
    TabIndex    = 78
    Top         = 4680
    Value       = 1  'Checked
    Width       = 615
  End
  Begin VB.CheckBox Check4
    Caption     = "04"
    BeginProperty Font
      name      = "MS SystemEx"
      charset   = 0
      weight    = 700
      size      = 7.5
      underline = 0  'False
      italic    = 0  'False
      strikethrough = 0  'False
    EndProperty
    Height      = 255
    Left        = 4440
    TabIndex    = 77
    Top         = 4440
    Value       = 1  'Checked
    Width       = 615
  End
  Begin VB.CheckBox Check3
    Caption     = "03"
    BeginProperty Font
      name      = "MS SystemEx"
      charset   = 0
      weight    = 700
      size      = 7.5
      underline = 0  'False
      italic    = 0  'False
      strikethrough = 0  'False
    EndProperty
    Height      = 255
    Left        = 4440
    TabIndex    = 76
    Top         = 4200
    Value       = 1  'Checked
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    Width      = 615
  End
  Begin VB.CheckBox Check2
    Caption    = "02"
    BeginProperty Font
      name     = "MS SystemEx"
      charset  = 0
      weight   = 700
      size     = 7.5
      underline  = 0  'False
      italic     = 0  'False
      strikethrough = 0  'False
    EndProperty
    Height     = 255
    Left       = 4440
    TabIndex   = 75
    Top        = 3960
    Value      = 1  'Checked
    Width      = 615
  End
  Begin VB.CheckBox Check1
    Caption    = "01"
    BeginProperty Font
      name     = "MS SystemEx"
      charset  = 0
      weight   = 700
      size     = 7.5
      underline  = 0  'False
      italic     = 0  'False
      strikethrough = 0  'False
    EndProperty
    Height     = 255
    Left       = 4440
    TabIndex   = 74
    Top        = 3720
    Value      = 1  'Checked
    Width      = 615
  End
  Begin VB.TextBox Text4
    Alignment  = 1  'Right Justify
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    BeginProperty Font
      name        = "MS SystemEx"
      charset     = 0
      weight      = 700
      size        = 7.5
      underline   = 0  'False
      italic      = 0  'False
      strikethrough = 0  'False
    EndProperty
    Height      = 270
    Left        = 8400
    TabIndex    = 60
    Text        = "W"
    Top         = 4440
    Width       = 375
  End
  Begin VB.VScrollBar VScroll1
    Height      = 255
    Left        = 6720
    Max         = 1
    Min         = 6
    TabIndex    = 69
    Top         = 4680
    Value       = 1
    Width       = 135
  End
  Begin VB.TextBox TxtStatus
    BeginProperty Font
      name        = "MS SystemEx"
      charset     = 0
      weight      = 700
      size        = 7.5
      underline   = 0  'False
      italic      = 0  'False
      strikethrough = 0  'False
    EndProperty
    Height      = 270
    Index       = 6
    Left        = 5040
    TabIndex    = 68
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    Text        = "OnL"
    Top         = 4920
    Width       = 615
  End
  Begin VB.TextBox TxtStatus
    BeginProperty Font
      name        = "MS SystemEx"
      charset     = 0
      weight      = 700
      size        = 7.5
      underline   = 0   'False
      italic      = 0   'False
      strikethrough = 0   'False
    EndProperty
    Height      = 270
    Index       = 5
    Left        = 5040
    TabIndex    = 67
    Text        = "OnL"
    Top         = 4680
    Width       = 615
  End
  Begin VB.TextBox TxtStatus
    BeginProperty Font
      name        = "MS SystemEx"
      charset     = 0
      weight      = 700
      size        = 7.5
      underline   = 0   'False
      italic      = 0   'False
      strikethrough = 0   'False
    EndProperty
    Height      = 270
    Index       = 4
    Left        = 5040
    TabIndex    = 66
    Text        = "OnL"
    Top         = 4440
    Width       = 615
  End
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Begin VB.TextBox TxtStatus
  BeginProperty Font
    name        = "MS SystemEx"
    charset     = 0
    weight      = 700
    size        = 7.5
    underline   = 0  'False
    italic      = 0  'False
    strikethrough = 0  'False
  EndProperty
  Height      = 270
  Index       = 3
  Left        = 5040
  TabIndex    = 65
  Text        = "OnL"
  Top         = 4200
  Width       = 615
End
Begin VB.TextBox TxtStatus
  BeginProperty Font
    name        = "MS SystemEx"
    charset     = 0
    weight      = 700
    size        = 7.5
    underline   = 0  'False
    italic      = 0  'False
    strikethrough = 0  'False
  EndProperty
  Height      = 270
  Index       = 2
  Left        = 5040
  TabIndex    = 64
  Text        = "OnL"
  Top         = 3960
  Width       = 615
End
Begin VB.PictureBox PicPV_Graph
  AutoRedraw  = -1  'True
  BackColor   = &H00000000&
  FillColor   = &H0000FF00&
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    FillStyle     = 0  'Solid
    BeginProperty Font
      name        = "MS LineDraw"
      charset     = 2
      weight      = 700
      size        = 8.25
      underline   = 0  'False
      italic      = 0  'False
      strikethrough = 0  'False
    EndProperty
    ForeColor     = &H00000080&
    Height        = 1000
    Left          = 4680
    ScaleHeight   = 100
    ScaleMode     = 0  'User
    ScaleWidth    = 196
    TabIndex      = 17
    Top           = 360
    Width         = 3000
  End
  Begin VB.PictureBox PicMV_Graph
    AutoRedraw    = -1  'True
    BackColor     = &H00000000&
    FillColor     = &H0000FF00&
    FillStyle     = 0  'Solid
    BeginProperty Font
      name        = "MS LineDraw"
      charset     = 2
      weight      = 700
      size        = 8.25
      underline   = 0  'False
      italic      = 0  'False
      strikethrough = 0  'False
    EndProperty
    ForeColor     = &H00000080&
    Height        = 1000
    Left          = 4680
    ScaleHeight   = 647.26
    ScaleMode     = 0  'User
    ScaleWidth    = 1318.386
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    TabIndex      = 19
    Top           = 1800
    Width         = 3000
  End
  Begin VB.TextBox Text3
    Alignment     = 1  'Right Justify
    BeginProperty Font
      name        = "MS SystemEx"
      charset     = 0
      weight      = 700
      size        = 7.5
      underline   = 0  'False
      italic      = 0  'False
      strikethrough = 0  'False
    EndProperty
    Height        = 270
    Left          = 6360
    TabIndex      = 58
    Text          = "01"
    Top           = 4680
    Width         = 375
  End
  Begin VB.TextBox Text00SetPoint
    BeginProperty Font
      name        = "MS SystemEx"
      charset     = 0
      weight      = 700
      size        = 7.5
      underline   = 0  'False
      italic      = 0  'False
      strikethrough = 0  'False
    EndProperty
    Height        = 270
    Left          = 240
    TabIndex      = 37
    Text          = "50"
    Top           = 3840
    Width         = 735
  End
  Begin VB.TextBox Text00MV
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    BackColor      = &H00C0FFFF&
    Enabled        = 0  'False
    BeginProperty Font
      name         = "MS SystemEx"
      charset      = 0
      weight       = 700
      size         = 7.5
      underline    = 0  'False
      italic       = 0  'False
      strikethrough = 0  'False
    EndProperty
    Height         = 270
    Left           = 3000
    TabIndex       = 50
    Top            = 3360
    Width          = 615
  End
  Begin VB.TextBox Text00LowAlarm
    BeginProperty Font
      name         = "MS SystemEx"
      charset      = 0
      weight       = 700
      size         = 7.5
      underline    = 0  'False
      italic       = 0  'False
      strikethrough = 0  'False
    EndProperty
    Height         = 270
    Left           = 3000
    TabIndex       = 42
    Text           = "10"
    Top            = 4680
    Width          = 615
  End
  Begin VB.TextBox Text00Kp
    BeginProperty Font
      name         = "MS SystemEx"
      charset      = 0
      weight       = 700
      size         = 7.5
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
      underline    = 0  'False
      italic       = 0  'False
      strikethrough = 0  'False
    EndProperty
    Height       = 270
    Left         = 3000
    TabIndex     = 36
    Text         = ".15"
    Top          = 3600
    Width        = 615
  End
  Begin VB.TextBox Text00Ki
    BeginProperty Font
      name         = "MS SystemEx"
      charset      = 0
      weight       = 700
      size         = 7.5
      underline    = 0  'False
      italic       = 0  'False
      strikethrough = 0  'False
    EndProperty
    Height       = 270
    Left         = 3000
    TabIndex     = 39
    Text         = ".25"
    Top          = 3840
    Width        = 615
  End
  Begin VB.TextBox Text00Kd
    BeginProperty Font
      name         = "MS SystemEx"
      charset      = 0
      weight       = 700
      size         = 7.5
      underline    = 0  'False
      italic       = 0  'False
      strikethrough = 0  'False
    EndProperty
    Height       = 270
    Left         = 3000
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    TabIndex      = 38
    Text          = ".05"
    Top           = 4080
    Width         = 615
  End
  Begin VB.TextBox Text00HighAlarm
    BeginProperty Font
      name        = "MS SystemEx"
      charset     = 0
      weight      = 700
      size        = 7.5
      underline   = 0   'False
      italic      = 0   'False
      strikethrough = 0   'False
    EndProperty
    Height        = 270
    Left          = 3000
    TabIndex      = 41
    Text          = "90"
    Top           = 4920
    Width         = 615
  End
  Begin VB.TextBox Text00Bias
    BeginProperty Font
      name        = "MS SystemEx"
      charset     = 0
      weight      = 700
      size        = 7.5
      underline   = 0   'False
      italic      = 0   'False
      strikethrough = 0   'False
    EndProperty
    Height        = 270
    Left          = 3000
    TabIndex      = 40
    Text          = "50.2"
    Top           = 4320
    Width         = 615
  End
  Begin VB.CommandButton CommandStart
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    Caption     = "START"
    BeginProperty Font
      name      = "Lucida Sans"
      charset   = 0
      weight    = 700
      size      = 8.25
      underline = 0  'False
      italic    = -1 'True
      strikethrough = 0  'False
    EndProperty
    Height      = 255
    Left        = 720
    TabIndex    = 5
    Top         = 1680
    Width       = 1215
  End
  Begin VB.TextBox Text2
    BeginProperty Font
      name      = "MS Sans Serif"
      charset   = 222
      weight    = 400
      size      = 8.25
      underline = 0  'False
      italic    = 0  'False
      strikethrough = 0  'False
    EndProperty
    Height      = 315
    Left        = 2280
    TabIndex    = 4
    Top         = 2280
    Width       = 1575
  End
  Begin VB.TextBox Text1
    Alignment   = 1  'Right Justify
    BeginProperty Font
      name      = "MS Sans Serif"
      charset   = 222
      weight    = 400
      size      = 8.25
      underline = 0  'False
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    italic      = 0  False
    strikethrough  = 0  False
  EndProperty
  Height      = 315
  Left        = 480
  TabIndex    = 1
  Top         = 2280
  Width       = 1575
End
Begin GaugeLib.Gauge GaugeMV
  Height      = 975
  Left        = 8790
  TabIndex    = 12
  Top         = 1830
  Width       = 135
  _Version    = 65536
  _ExtentX    = 238
  _ExtentY    = 1720
  _StockProps  = 73
  ForeColor   = 255
  BackColor   = -2147483644
  InnerTop    = 3
  InnerLeft   = 3
  InnerRight  = 3
  InnerBottom  = 3
  Max         = 255
  Value       = 80
  Style       = 1
  NeedleWidth  = 1
End
Begin GaugeLib.Gauge GaugePV
  Height      = 990
  Left        = 8790
  TabIndex    = 11
  Top         = 390
  Width       = 135
  _Version    = 65536
  _ExtentX    = 238
  _ExtentY    = 1746
  _StockProps  = 73
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    ForeColor     = 16776960
    BackColor     = -2147483644
    InnerTop      = 3
    InnerLeft     = 3
    InnerRight    = 3
    InnerBottom   = 3
    Max           = 255
    Value         = 80
    Style         = 1
    Autosize      = -1  'True
    NeedleWidth   = 1
  End
  Begin VB.TextBox TxtStatus
    BeginProperty Font
      name        = "MS SystemEx"
      charset     = 0
      weight      = 700
      size        = 7.5
      underline   = 0   'False
      italic      = 0   'False
      strikethrough = 0 'False
    EndProperty
    Height        = 270
    Index         = 1
    Left          = 5040
    TabIndex      = 63
    Text          = "OnL"
    Top           = 3720
    Width         = 615
  End
  Begin VB.TextBox N04txt
    BeginProperty Font
      name        = "MS SystemEx"
      charset     = 0
      weight      = 700
      size        = 7.5
      underline   = 0   'False
      italic      = 0   'False
      strikethrough = 0 'False
    EndProperty
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    Height      = 270
    Left        = 6480
    TabIndex    = 82
    Text        = "00.00"
    Top         = 720
    Width       = 735
  End
  Begin VB.TextBox Text6
    BeginProperty Font
      name        = "MS SystemEx"
      charset     = 0
      weight      = 700
      size        = 7.5
      underline   = 0   'False
      italic      = 0   'False
      strikethrough = 0   'False
    EndProperty
    Height      = 270
    Left        = 6960
    TabIndex    = 89
    Text        = "00"
    Top         = 1680
    Width       = 615
  End
  Begin VB.TextBox Text5
    BeginProperty Font
      name        = "MS SystemEx"
      charset     = 0
      weight      = 700
      size        = 7.5
      underline   = 0   'False
      italic      = 0   'False
      strikethrough = 0   'False
    EndProperty
    Height      = 270
    Left        = 6960
    TabIndex    = 88
    Text        = "00"
    Top         = 960
    Width       = 615
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
End
Begin VB.Label Label39
  Alignment     = 2  'Center
  BackStyle     = 0  'Transparent
  BorderStyle   = 1  'Fixed Single
  Caption       = "Hex"
  BeginProperty Font
    name        = "Lucida Sans"
    charset     = 0
    weight      = 700
    size        = 8.25
    underline   = 0  'False
    italic      = -1 'True
    strikethrough = 0  'False
  EndProperty
  Height        = 255
  Left          = 7560
  TabIndex      = 93
  Top           = 1680
  Width         = 495
End
Begin VB.Label Label38
  Alignment     = 2  'Center
  BackStyle     = 0  'Transparent
  BorderStyle   = 1  'Fixed Single
  Caption       = "Hex"
  BeginProperty Font
    name        = "Lucida Sans"
    charset     = 0
    weight      = 700
    size        = 8.25
    underline   = 0  'False
    italic      = -1 'True
    strikethrough = 0  'False
  EndProperty
  Height        = 255
  Left          = 7560
  TabIndex      = 92
  Top           = 960
  Width         = 495
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
End
Begin VB.Label Label41
  Alignment     = 2  'Center
  BackStyle     = 0  'Transparent
  BorderStyle   = 1  'Fixed Single
  Caption       = "ST.No. 05 Input"
  BeginProperty Font
    name        = "Lucida Sans"
    charset     = 0
    weight      = 700
    size        = 8.25
    underline   = 0  'False
    italic      = -1 'True
    strikethrough = 0  'False
  EndProperty
  Height        = 255
  Left          = 5160
  TabIndex      = 91
  Top           = 960
  Width         = 1815
End
Begin VB.Label Label40
  Alignment     = 2  'Center
  BackStyle     = 0  'Transparent
  BorderStyle   = 1  'Fixed Single
  Caption       = "ST.No. 06 Output"
  BeginProperty Font
    name        = "Lucida Sans"
    charset     = 0
    weight      = 700
    size        = 8.25
    underline   = 0  'False
    italic      = -1 'True
    strikethrough = 0  'False
  EndProperty
  Height        = 255
  Left          = 5160
  TabIndex      = 90
  Top           = 1680
  Width         = 1815
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
End
Begin VB.Line Line18
  X1        = 4320
  X2        = 4440
  Y1        = 5100
  Y2        = 5100
End
Begin VB.Line Line17
  X1        = 4320
  X2        = 4440
  Y1        = 4740
  Y2        = 4740
End
Begin VB.Line Line16
  X1        = 4320
  X2        = 4320
  Y1        = 5100
  Y2        = 4740
End
Begin VB.Line Line15
  X1        = 4320
  X2        = 4440
  Y1        = 4260
  Y2        = 4260
End
Begin VB.Line Line12
  X1        = 4320
  X2        = 4320
  Y1        = 4620
  Y2        = 4260
End
Begin VB.Line Line11
  X1        = 4320
  X2        = 4440
  Y1        = 4620
  Y2        = 4620
End
Begin VB.Label Label37
  Alignment    = 2 'Center
  BackStyle    = 0 'Transparent
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    BorderStyle    =  1  'Fixed Single
    Caption        =  "ST.No. 03 PV"
    BeginProperty Font
      name         =  "Lucida Sans"
      charset      =  0
      weight       =  700
      size         =  8.25
      underline    =  0  'False
      italic       =  -1  'True
      strikethrough  =  0  'False
    EndProperty
    Height         =  255
    Left           =  4680
    TabIndex       =  87
    Top            =  2160
    Width          =  3015
  End
  Begin VB.Label Label35
    Alignment      =  2  'Center
    BackStyle      =  0  'Transparent
    BorderStyle    =  1  'Fixed Single
    Caption        =  "%"
    BeginProperty Font
      name         =  "Lucida Sans"
      charset      =  0
      weight       =  700
      size         =  8.25
      underline    =  0  'False
      italic       =  -1  'True
      strikethrough  =  0  'False
    EndProperty
    Height         =  255
    Left           =  7200
    TabIndex       =  83
    Top            =  720
    Width          =  495
  End
  Begin VB.Line Line14
    BorderColor    =  &H00C0C0C0&
    X1             =  7200
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  X2          = 7200
  Y1          = 5280
  Y2          = 5880
End
Begin VB.Line Line13
  BorderColor   = &H00C0C0C0&
  X1          = 4440
  X2          = 4440
  Y1          = 5280
  Y2          = 5880
End
Begin VB.Label Label34
  Caption       = " Technology Ladkrabang"
  BeginProperty Font
    name        = "MS SystemEx"
    charset     = 0
    weight      = 700
    size        = 7.5
    underline   = 0  'False
    italic      = 0  'False
    strikethrough = 0  'False
  EndProperty
  Height        = 255
  Index         = 8
  Left          = 4560
  TabIndex      = 81
  Top           = 5640
  Width         = 2535
End
Begin VB.Label Label34
  Caption       = "King Mongkut's Institute of"
  BeginProperty Font
    name        = "MS SystemEx"
    charset     = 0
    weight      = 700
    size        = 7.5
    underline   = 0  'False
    italic      = 0  'False
    strikethrough = 0  'False
  EndProperty
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    Height      = 255
    Index       = 5
    Left        = 4560
    TabIndex    = 80
    Top         = 5400
    Width       = 2655
  End
  Begin VB.Label Label34
    Caption     = "General Instrument Protocol"
    BeginProperty Font
      name        = "MS SystemEx"
      charset     = 0
      weight      = 700
      size        = 7.5
      underline   = 0   'False
      italic      = 0   'False
      strikethrough = 0   'False
    EndProperty
    Height      = 255
    Index       = 2
    Left        = 6120
    TabIndex    = 62
    Top         = 3240
    Width       = 2775
  End
  Begin VB.Label cTime
    Alignment   = 1  'Right Justify
    Caption     = "CTime (S) = 0"
    BeginProperty Font
      name        = "MS SystemEx"
      charset     = 0
      weight      = 700
      size        = 7.5
      underline   = 0   'False
      italic      = 0   'False
      strikethrough = 0   'False
    EndProperty
    Height      = 255
    Left        = 7200
    TabIndex    = 73
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    Top         = 5400
    Width       = 1815
  End
  Begin VB.Label Label34
    Alignment   = 2 'Center
    BackColor   = &H00808080&
    BorderStyle = 1 'Fixed Single
    Caption     = "--- GCS ---"
    BeginProperty Font
      name      = "MS SystemEx"
      charset   = 0
      weight    = 700
      size      = 7.5
      underline = 0 'False
      italic    = 0 'False
      strikethrough = 0 'False
    EndProperty
    Height      = 255
    Index       = 4
    Left        = 7080
    TabIndex    = 61
    Top         = 4440
    Width       = 1335
  End
  Begin VB.Label Label34
    BorderStyle = 1 'Fixed Single
    Caption     = "Stat."
    BeginProperty Font
      name      = "MS SystemEx"
      charset   = 0
      weight    = 700
      size      = 7.5
      underline = 0 'False
      italic    = 0 'False
      strikethrough = 0 'False
    EndProperty
    Height      = 255
    Index       = 6
    Left        = 5160
    TabIndex    = 70
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    Top        = 3360
    Width      = 495
  End
  Begin VB.Label Label34
    BorderStyle  = 1  'Fixed Single
    Caption      = "ST.No."
    BeginProperty Font
      name       = "MS SystemEx"
      charset    = 0
      weight     = 700
      size       = 7.5
      underline  = 0  'False
      italic     = 0  'False
      strikethrough = 0  'False
    EndProperty
    Height     = 255
    Index      = 7
    Left       = 4320
    TabIndex   = 71
    Top        = 3360
    Width      = 735
  End
  Begin Threed.SSCheck NOv
    Height     = 375
    Left       = 2160
    TabIndex   = 72
    Top        = 5400
    Width      = 2175
    _Version   = 65536
    _ExtentX   = 3836
    _ExtentY   = 661
    _StockProps  = 78
    Caption    = "NORMAL OPEN VALVE"
    BeginProperty Font {0BE35203-8F91-11CE-9DE3-00AA004BB851}
      name       = "Lucida Sans"
      charset    = 0
      weight     = 700
      size       = 7.5
      underline  = 0  'False
      italic     = -1  'True
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    strikethrough = 0 'False
  EndProperty
  Value      = -1 'True
End
Begin VB.Label Label34
  Alignment    = 2 'Center
  BackStyle    = 0 'Transparent
  Caption      = "ST.No."
  BeginProperty Font
    name       = "MS SystemEx"
    charset    = 0
    weight     = 700
    size       = 7.5
    underline  = 0 'False
    italic     = 0 'False
    strikethrough = 0 'False
  EndProperty
  Height       = 255
  Index        = 3
  Left         = 6240
  TabIndex     = 59
  Top          = 4440
  Width        = 735
End
Begin VB.Label Label34
  Caption      = "MV_Graph"
  BeginProperty Font
    name       = "MS SystemEx"
    charset    = 0
    weight     = 700
    size       = 7.5
    underline  = 0 'False
    italic     = 0 'False
    strikethrough = 0 'False
  EndProperty
  Height       = 255
  Index        = 1
  Left         = 4680
  TabIndex     = 57
  Top          = 1560
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    Width      = 1095
  End
  Begin VB.Label Label34
    Caption    = "PV_Graph"
    BeginProperty Font
      name     = "MS SystemEx"
      charset  = 0
      weight   = 700
      size     = 7.5
      underline  = 0  'False
      italic     = 0  'False
      strikethrough  = 0  'False
    EndProperty
    Height     = 255
    Index      = 0
    Left       = 4680
    TabIndex   = 56
    Top        = 120
    Width      = 1095
  End
  Begin VB.Label ErrCount
    Alignment  = 1  'Right Justify
    Caption    = "DATAErr = 0"
    BeginProperty Font
      name     = "MS SystemEx"
      charset  = 0
      weight   = 700
      size     = 7.5
      underline  = 0  'False
      italic     = 0  'False
      strikethrough  = 0  'False
    EndProperty
    Height     = 255
    Left       = 7320
    TabIndex   = 55
    Top        = 5640
    Width      = 1695
  End
  Begin VB.Label Label33
    Caption    = "TimeX"
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    BeginProperty Font
      name        = "Courier New"
      charset     = 0
      weight      = 400
      size        = 8.25
      underline   = 0  'False
      italic      = 0  'False
      strikethrough = 0  'False
    EndProperty
    Height      = 255
    Left        = 8520
    TabIndex    = 54
    Top         = 1440
    Width       = 615
  End
  Begin ComctlLib.Slider SliderTimeX
    Height      = 255
    Left        = 7800
    TabIndex    = 53
    Top         = 1440
    Width       = 735
    _Version    = 65536
    _ExtentX    = 1296
    _ExtentY    = 450
    _StockProps = 64
    Max         = 100
    Min         = 1
    SelectRange = -1  'True
    SelStart    = 10
    SelLength   = 30
    TickFrequency = 10
    Value       = 10
  End
  Begin VB.Label Label32
    Alignment   = 2  'Center
    BackStyle   = 0  'Transparent
    BorderStyle = 1  'Fixed Single
    Caption     = "%"
    BeginProperty Font
      name        = "Lucida Sans"
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    charset      = 0
    weight       = 700
    size         = 8.25
    underline    = 0  'False
    italic       = -1 'True
    strikethrough = 0 'False
  EndProperty
  Height       = 255
  Left         = 960
  TabIndex     = 52
  Top          = 3840
  Width        = 375
End
Begin VB.Label Label31
  Alignment    = 2 'Center
  BackStyle    = 0 'Transparent
  BorderStyle  = 1 'Fixed Single
  Caption      = "%"
  BeginProperty Font
    name       = "Lucida Sans"
    charset    = 0
    weight     = 700
    size       = 8.25
    underline  = 0 'False
    italic     = -1 'True
    strikethrough = 0 'False
  EndProperty
  Height       = 255
  Left         = 3600
  TabIndex     = 51
  Top          = 3360
  Width        = 495
End
Begin VB.Label Label30
  Alignment    = 2 'Center
  BorderStyle  = 1 'Fixed Single
  Caption      = "MV"
  BeginProperty Font
    name       = "Lucida Sans"
    charset    = 0
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    weight      = 700
    size        = 8.25
    underline   = 0  'False
    italic      = -1 'True
    strikethrough = 0  'False
  EndProperty
  Height      = 255
  Left        = 1440
  TabIndex    = 49
  Top         = 3360
  Width       = 1575
End
Begin VB.Label Label29
  Alignment   = 2  'Center
  BackStyle   = 0  'Transparent
  BorderStyle = 1  'Fixed Single
  Caption     = "%"
  BeginProperty Font
    name      = "Lucida Sans"
    charset   = 0
    weight    = 700
    size      = 8.25
    underline = 0  'False
    italic    = -1 'True
    strikethrough = 0  'False
  EndProperty
  Height      = 255
  Left        = 3600
  TabIndex    = 48
  Top         = 4920
  Width       = 495
End
Begin VB.Label Label28
  Alignment   = 2  'Center
  BackStyle   = 0  'Transparent
  BorderStyle = 1  'Fixed Single
  Caption     = "%"
  BeginProperty Font
    name      = "Lucida Sans"
    charset   = 0
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    weight      = 700
    size        = 8.25
    underline   = 0  'False
    italic      = -1 'True
    strikethrough = 0  'False
  EndProperty
  Height      = 255
  Left        = 3600
  TabIndex    = 47
  Top         = 4680
  Width       = 495
End
Begin VB.Label Label27
  Alignment   = 2 'Center
  BackStyle   = 0 'Transparent
  BorderStyle = 1 'Fixed Single
  Caption     = "%"
  BeginProperty Font
    name      = "Lucida Sans"
    charset   = 0
    weight    = 700
    size      = 8.25
    underline = 0  'False
    italic    = -1 'True
    strikethrough = 0  'False
  EndProperty
  Height      = 255
  Left        = 3600
  TabIndex    = 46
  Top         = 4320
  Width       = 495
End
Begin VB.Label Label26
  Alignment   = 2 'Center
  BackStyle   = 0 'Transparent
  BorderStyle = 1 'Fixed Single
  BeginProperty Font
    name      = "Lucida Sans"
    charset   = 0
    weight    = 700
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    size        = 8.25
    underline   = 0  'False
    italic      = -1 'True
    strikethrough = 0  'False
  EndProperty
  Height      = 255
  Left        = 3600
  TabIndex    = 45
  Top         = 4080
  Width       = 495
End
Begin VB.Label Label25
  Alignment   = 2 'Center
  BackStyle   = 0 'Transparent
  BorderStyle = 1 'Fixed Single
  BeginProperty Font
    name        = "Lucida Sans"
    charset     = 0
    weight      = 700
    size        = 8.25
    underline   = 0  'False
    italic      = -1 'True
    strikethrough = 0  'False
  EndProperty
  Height      = 255
  Left        = 3600
  TabIndex    = 44
  Top         = 3840
  Width       = 495
End
Begin VB.Label Label24
  Alignment   = 2 'Center
  BackStyle   = 0 'Transparent
  BorderStyle = 1 'Fixed Single
  BeginProperty Font
    name        = "Lucida Sans"
    charset     = 0
    weight      = 700
    size        = 8.25
    underline   = 0  'False
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    italic      = -1  'True
    strikethrough = 0  'False
  EndProperty
  Height      = 255
  Left        = 3600
  TabIndex    = 43
  Top         = 3600
  Width       = 495
End
Begin VB.Label Label23
  Alignment   = 2  'Center
  BackStyle   = 0  'Transparent
  BorderStyle = 1  'Fixed Single
  Caption     = "PV LOW ALARM POINT"
  BeginProperty Font
    name      = "Lucida Sans"
    charset   = 0
    weight    = 700
    size      = 8.25
    underline = 0  'False
    italic    = -1  'True
    strikethrough = 0  'False
  EndProperty
  Height      = 255
  Left        = 240
  TabIndex    = 33
  Top         = 4680
  Width       = 2775
End
Begin VB.Label Label22
  Alignment   = 2  'Center
  BackStyle   = 0  'Transparent
  BorderStyle = 1  'Fixed Single
  Caption     = "PV HIGH ALARM POINT"
  BeginProperty Font
    name      = "Lucida Sans"
    charset   = 0
    weight    = 700
    size      = 8.25
    underline = 0  'False
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    italic      = -1  'True
    strikethrough = 0  'False
  EndProperty
  Height      = 255
  Left        = 240
  TabIndex    = 34
  Top         = 4920
  Width       = 2775
End
Begin VB.Label Label18
  Alignment   = 2  'Center
  BackStyle   = 0  'Transparent
  BorderStyle = 1  'Fixed Single
  Caption     = "LOOP GAIN (Kp)"
  BeginProperty Font
    name      = "Lucida Sans"
    charset   = 0
    weight    = 700
    size      = 8.25
    underline = 0  'False
    italic    = -1  'True
    strikethrough = 0  'False
  EndProperty
  Height      = 255
  Left        = 1440
  TabIndex    = 27
  Top         = 3600
  Width       = 1575
End
Begin VB.Label Label19
  Alignment   = 2  'Center
  BackStyle   = 0  'Transparent
  BorderStyle = 1  'Fixed Single
  Caption     = "Ki"
  BeginProperty Font
    name      = "Lucida Sans"
    charset   = 0
    weight    = 700
    size      = 8.25
    underline = 0  'False
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    italic      = -1  'True
    strikethrough = 0  'False
  EndProperty
  Height      = 255
  Left        = 1440
  TabIndex    = 28
  Top         = 3840
  Width       = 1575
End
Begin VB.Label Label20
  Alignment   = 2  'Center
  BackStyle   = 0  'Transparent
  BorderStyle = 1  'Fixed Single
  Caption     = "Kd"
  BeginProperty Font
    name      = "Lucida Sans"
    charset   = 0
    weight    = 700
    size      = 8.25
    underline = 0  'False
    italic    = -1  'True
    strikethrough = 0  'False
  EndProperty
  Height      = 255
  Left        = 1440
  TabIndex    = 29
  Top         = 4080
  Width       = 1575
End
Begin Threed.SSCheck SSChk00Rev
  Height      = 375
  Left        = 240
  TabIndex    = 35
  Top         = 5400
  Width       = 2175
  _Version    = 65536
  _ExtentX    = 3836
  _ExtentY    = 661
  _StockProps = 78
  Caption     = "REVERSE ACTING"
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    BeginProperty Font {0BE35203-8F91-11CE-9DE3-00AA004BB851}
      name        = "Lucida Sans"
      charset     = 0
      weight      = 700
      size        = 7.5
      underline   = 0  'False
      italic      = -1  'True
      strikethrough = 0  'False
    EndProperty
    Value       = -1  'True
  End
  Begin Threed.SSOption SSOpt00Manual
    Height      = 375
    Left        = 1440
    TabIndex    = 32
    Top         = 2880
    Width       = 1095
    _Version    = 65536
    _ExtentX    = 1931
    _ExtentY    = 661
    _StockProps = 78
    Caption     = "MANUAL"
    BeginProperty Font {0BE35203-8F91-11CE-9DE3-00AA004BB851}
      name        = "Lucida Sans"
      charset     = 0
      weight      = 700
      size        = 7.5
      underline   = 0  'False
      italic      = -1  'True
      strikethrough = 0  'False
    EndProperty
    Font3D      = 1
  End
  Begin Threed.SSOption SSOpt00Auto
    Height      = 375
    Left        = 240
    TabIndex    = 31
    Top         = 2880
    Width       = 855
    _Version    = 65536
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    _ExtentX      = 1508
    _ExtentY      = 661
    _StockProps   = 78
    Caption       = "AUTO"
    BeginProperty Font {0BE35203-8F91-11CE-9DE3-00AA004BB851}
      name        = "Lucida Sans"
      charset     = 0
      weight      = 700
      size        = 7.5
      underline   = 0  'False
      italic      = -1 'True
      strikethrough = 0  'False
    EndProperty
    Value         = -1 'True
    Font3D        = 1
  End
  Begin VB.Label Label21
    Alignment     = 2 'Center
    BackStyle     = 0 'Transparent
    BorderStyle   = 1 'Fixed Single
    Caption       = "BIAS"
    BeginProperty Font
      name        = "Lucida Sans"
      charset     = 0
      weight      = 700
      size        = 8.25
      underline   = 0  'False
      italic      = -1 'True
      strikethrough = 0  'False
    EndProperty
    Height        = 255
    Left          = 1440
    TabIndex      = 30
    Top           = 4320
    Width         = 1575
  End
  Begin ComctlLib.Slider Slider00SetPoint
    Height        = 270
    Left          = 240
    TabIndex      = 26
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    Top         = 4200
    Width       = 1095
    _Version    = 65536
    _ExtentX    = 1931
    _ExtentY    = 476
    _StockProps = 64
    Max         = 100
    TickFrequency = 10
  End
  Begin VB.Line Line10
    X1          = 4680
    X2          = 4680
    Y1          = 2880
    Y2          = 2820
  End
  Begin VB.Label Label7
    Alignment   = 2 'Center
    BorderStyle = 1 'Fixed Single
    Caption     = "MV"
    BeginProperty Font
      name      = "Lucida Sans"
      charset   = 0
      weight    = 700
      size      = 8.25
      underline = 0 'False
      italic    = -1 'True
      strikethrough = 0 'False
    EndProperty
    Height      = 255
    Left        = 7680
    TabIndex    = 13
    Top         = 1800
    Width       = 975
  End
  Begin VB.Label Label9
    Alignment   = 1 'Right Justify
    BorderStyle = 1 'Fixed Single
    BeginProperty Font
      name      = "MS Sans Serif"
      charset   = 222
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    weight      = 700
    size        = 8.25
    underline   = 0  'False
    italic      = 0  'False
    strikethrough = 0  'False
  EndProperty
  Height      = 255
  Left        = 7680
  TabIndex    = 15
  Top         = 2040
  Width       = 975
End
Begin VB.Label Label8
  Alignment   = 1  'Right Justify
  BorderStyle = 1  'Fixed Single
  BeginProperty Font
    name      = "MS Sans Serif"
    charset   = 222
    weight    = 700
    size      = 8.25
    underline = 0  'False
    italic    = 0  'False
    strikethrough = 0  'False
  EndProperty
  Height      = 255
  Left        = 7680
  TabIndex    = 14
  Top         = 2280
  Width       = 975
End
Begin VB.Label Label17
  Alignment   = 2  'Center
  BackStyle   = 0  'Transparent
  Caption     = "t"
  BeginProperty Font
    name      = "Angsana New"
    charset   = 0
    weight    = 700
    size      = 12
    underline = 0  'False
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    italic      = 0   'False
    strikethrough = 0   'False
  EndProperty
  Height      = 255
  Left        = 7440
  TabIndex    = 25
  Top         = 1365
  Width       = 255
End
Begin VB.Label Label15
  Alignment   = 2   'Center
  BackStyle   = 0   'Transparent
  Caption     = "t"
  BeginProperty Font
    name      = "Angsana New"
    charset   = 0
    weight    = 700
    size      = 12
    underline = 0   'False
    italic    = 0   'False
    strikethrough = 0   'False
  EndProperty
  Height      = 255
  Left        = 7440
  TabIndex    = 24
  Top         = 2820
  Width       = 255
End
Begin VB.Label Label13
  Alignment   = 2   'Center
  BackStyle   = 0   'Transparent
  Caption     = "0"
  BeginProperty Font
    name      = "Angsana New"
    charset   = 0
    weight    = 700
    size      = 9.75
    underline = 0   'False
    italic    = 0   'False
    strikethrough = 0   'False
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    EndProperty
    Height      = 255
    Left        = 4320
    TabIndex    = 23
    Top         = 2610
    Width       = 255
  End
  Begin VB.Label Label12
    Alignment   = 2 'Center
    BackStyle   = 0 'Transparent
    Caption     = "100%"
    BeginProperty Font
      name      = "Angsana New"
      charset   = 0
      weight    = 700
      size      = 9.75
      underline = 0 'False
      italic    = 0 'False
      strikethrough = 0 'False
    EndProperty
    Height      = 255
    Left        = 4080
    TabIndex    = 22
    Top         = 1680
    Width       = 615
  End
  Begin VB.Line Line9
    X1          = 4560
    X2          = 4620
    Y1          = 1800
    Y2          = 1800
  End
  Begin VB.Line Line8
    X1          = 4560
    X2          = 4620
    Y1          = 2760
    Y2          = 2760
  End
  Begin VB.Line Line3
    X1          = 4560
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    X2          = 4560
    Y1          = 1800
    Y2          = 2760
  End
  Begin VB.Label Label16
    Alignment   = 2 'Center
    BackStyle   = 0 'Transparent
    Caption     = "0"
    BeginProperty Font
      name      = "Angsana New"
      charset   = 0
      weight    = 700
      size      = 9.75
      underline = 0 'False
      italic    = 0 'False
      strikethrough = 0 'False
    EndProperty
    Height      = 255
    Left        = 4320
    TabIndex    = 21
    Top         = 1200
    Width       = 255
  End
  Begin VB.Label Label14
    Alignment   = 2 'Center
    BackStyle   = 0 'Transparent
    Caption     = "100%"
    BeginProperty Font
      name      = "Angsana New"
      charset   = 0
      weight    = 700
      size      = 9.75
      underline = 0 'False
      italic    = 0 'False
      strikethrough = 0 'False
    EndProperty
    Height      = 255
    Left        = 4080
    TabIndex    = 20
    Top         = 240
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    Width      = 615
  End
  Begin VB.Line Line7
    X1         = 4680
    X2         = 4680
    Y1         = 1435
    Y2         = 1365
  End
  Begin VB.Line Line6
    X1         = 4560
    X2         = 4620
    Y1         = 360
    Y2         = 360
  End
  Begin VB.Line Line5
    X1         = 4560
    X2         = 4620
    Y1         = 1320
    Y2         = 1320
  End
  Begin VB.Label Label6
    Alignment      = 2  'Center
    BorderStyle    = 1  'Fixed Single
    Caption        = "PV"
    BeginProperty Font
      name         = "Lucida Sans"
      charset      = 0
      weight       = 700
      size         = 8.25
      underline    = 0   'False
      italic       = -1  'True
      strikethrough  = 0   'False
    EndProperty
    Height       = 255
    Left         = 7680
    TabIndex     = 10
    Top          = 360
    Width        = 975
  End
  Begin VB.Label Label3
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    Alignment      = 1  'Right Justify
    BorderStyle    = 1  'Fixed Single
    BeginProperty Font
      name         = "MS Sans Serif"
      charset      = 222
      weight       = 700
      size         = 8.25
      underline    = 0  'False
      italic       = 0  'False
      strikethrough = 0  'False
    EndProperty
    Height         = 255
    Left           = 7680
    TabIndex       = 8
    Top            = 600
    Width          = 975
  End
  Begin VB.Label Label5
    Alignment      = 1  'Right Justify
    BorderStyle    = 1  'Fixed Single
    BeginProperty Font
      name         = "MS Sans Serif"
      charset      = 222
      weight       = 700
      size         = 8.25
      underline    = 0  'False
      italic       = 0  'False
      strikethrough = 0  'False
    EndProperty
    Height         = 255
    Left           = 7680
    TabIndex       = 9
    Top            = 840
    Width          = 975
  End
  Begin VB.Label Label4
    Alignment      = 1  'Right Justify
    BorderStyle    = 1  'Fixed Single
    BeginProperty Font
      name         = "MS Sans Serif"
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    charset      = 222
    weight       = 700
    size         = 8.25
    underline    = 0  'False
    italic       = 0  'False
    strikethrough = 0  'False
  EndProperty
  Height       = 255
  Left         = 7680
  TabIndex     = 7
  Top          = 1080
  Width        = 975
End
Begin VB.Label Label10
  Alignment    = 1  'Right Justify
  BorderStyle  = 1  'Fixed Single
  BeginProperty Font
    name       = "MS Sans Serif"
    charset    = 222
    weight     = 700
    size       = 8.25
    underline  = 0  'False
    italic     = 0  'False
    strikethrough = 0  'False
  EndProperty
  Height       = 255
  Left         = 7680
  TabIndex     = 16
  Top          = 2520
  Width        = 975
End
Begin VB.Line Line4
  X1           = 4680
  X2           = 7680
  Y1           = 2880
  Y2           = 2880
End
Begin VB.Line Line2
  X1           = 4680
  X2           = 7680
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    Y1          = 1440
    Y2          = 1440
  End
  Begin VB.Line Line1
    X1          = 4560
    X2          = 4560
    Y1          = 360
    Y2          = 1320
  End
  Begin VB.Label Label11
    Alignment   = 2 'Center
    AutoSize    = -1 'True
    BackStyle   = 0 'Transparent
    BorderStyle = 1 'Fixed Single
    Caption     = "SETPOINT"
    BeginProperty Font
      name      = "Lucida Sans"
      charset   = 0
      weight    = 700
      size      = 8.25
      underline = 0 'False
      italic    = -1 'True
      strikethrough = 0 'False
    EndProperty
    Height      = 270
    Left        = 240
    TabIndex    = 18
    Top         = 3360
    Width       = 1095
  End
  Begin VB.Label Label2
    Alignment   = 2 'Center
    BackStyle   = 0 'Transparent
    Caption     = "To Controller "
    BeginProperty Font
      name      = "MS Sans Serif"
      charset   = 222
      weight    = 700
      size      = 8.25
      underline = 0 'False
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    italic      = 0  'False
    strikethrough  = 0  'False
  EndProperty
  Height      = 255
  Left        = 480
  TabIndex    = 3
  Top         = 2040
  Width       = 1575
End
Begin VB.Label Label1
  Alignment   = 2  'Center
  BackStyle   = 0  'Transparent
  Caption     = "From Controller"
  BeginProperty Font
    name      = "MS Sans Serif"
    charset   = 222
    weight    = 700
    size      = 8.25
    underline   = 0  'False
    italic      = 0  'False
    strikethrough  = 0  'False
  EndProperty
  Height      = 255
  Left        = 2280
  TabIndex    = 2
  Top         = 2040
  Width       = 1575
End
Begin ComctlLib.StatusBar StatusBar1
  Align       = 2  'Align Bottom
  Height      = 270
  Left        = 0
  TabIndex    = 0
  Top         = 5940
  Width       = 9240
  _Version    = 65536
  _ExtentX    = 16298
  _ExtentY    = 476
  _StockProps   = 68
  BeginProperty Font {0BE35203-8F91-11CE-9DE3-00AA004BB851}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    name        = "Lucida Sans"
    charset     = 0
    weight      = 700
    size        = 8.25
    underline   = 0  'False
    italic      = -1  'True
    strikethrough = 0  'False
  EndProperty
  AlignSet    = -1  'True
  MouseIcon   = "DCSV6Form1.frx":0000
  Style       = 1
  SimpleText  = "Ready"
  il          = "DCSV6Form1.frx":0452
End
Begin MSCommLib.MSComm MSComm2
  Left        = 240
  Top         = 240
  _Version    = 65536
  _ExtentX    = 847
  _ExtentY    = 847
  _StockProps = 0
  CDTimeout   = 0
  CommPort    = 2
  CTSTimeout  = 0
  DSRTimeout  = 0
  DTREnable   = 0  'False
  Handshaking = 0
  InBufferSize = 128
  InputLen    = 0
  Interval    = 1000
  NullDiscard = 0  'False
  OutBufferSize = 128
  ParityReplace = "?"
  RThreshold  = 0
  RTSEnable   = 0  'False
  Settings    = "9600,n,8,1"
  SThreshold  = 0
End
Begin VB.OLE OLE1
  BackStyle   = 0  'Transparent
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    BorderStyle    = 0  'None
    Class          = "MSWordArt.2"
    Enabled        = 0  'False
    Height         = 1575
    Left           = 600
    OleObjectBlob  = "DCSV6Form1.frx":051F
    SizeMode       = 1  'Stretch
    TabIndex       = 6
    Top            = 120
    Width          = 3135
  End
  Begin VB.Image Image1
    BorderStyle    = 1  'Fixed Single
    Height         = 2535
    Left           = 240
    Picture        = "DCSV6Form1.frx":7D37
    Stretch        = -1  'True
    Top            = 240
    Width          = 3855
  End
  Begin VB.Label Label36
    Alignment      = 2  'Center
    BackStyle      = 0  'Transparent
    BorderStyle    = 1  'Fixed Single
    Caption        = "ST.No. 04 MV"
    BeginProperty Font
      name         = "Lucida Sans"
      charset      = 0
      weight       = 700
      size         = 8.25
      underline    = 0  'False
      italic       = -1  'True
      strikethrough = 0  'False
    EndProperty
    Height         = 255
    Left           = 4680
    TabIndex       = 84
    Top            = 720
    Width          = 1815
  End
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  Begin VB.Image Image2
    Height      = 1575
    Left        = 5880
    Picture     = "DCSV6Form1.frx":A719
    Stretch     = -1 'True
    Top         = 3600
    Width       = 3135
  End
End
Attribute VB_Name = "DCSForm1"
Attribute VB_Creatable = False
Attribute VB_Exposed = False
Option Explicit

'DefInt A-Z

Dim SFlag, RfshG As Boolean
'Dim IncX, CX, CY, XPos1, YPos1 ' Declare variables.
Dim Line1st As Boolean
Dim LXpos1, LXpos2, LYpos1, LYpos2, Xpos1, Xpos2, Ypos1, Ypos2, IncX
Dim Dummy
Dim IOstring$, LastOut$
Dim ANTInput As String
Dim ANTOutput As String

Dim Vo, Von_1, en, en_1, en_2 As Double

Dim xxx As Single 'time test

Dim ST_No, GCS, ChkS As String

Dim DataErr(1), LpN As Integer

Dim cB(6) As Integer

Dim TmC As Single
Sub CommuShare() '"!! Note!  Cycle Time for ST.No01 must = variable T in ST.No. 01(Vo) Equation
While SFlag = True
  TmC = Timer
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
If Check1.Value = 1 Then InterChange
'xxx = xxx + 1
'If xxx = 250 Then MsgBox "55 Sec"
Do While Timer - TmC < 0.1
   Dummy = DoEvents()
Loop
If Val(text3.Text) = 1 Then cTime.Caption = "CTime (S) = " & Format$(Timer - (TmC - 0.111), ".000")

If Check2.Value = 1 Then InterChg2
   Do While Timer - TmC < 0.21
   Dummy = DoEvents()
Loop
If Val(text3.Text) = 2 Then cTime.Caption = "CTime (S) = " & Format$(Timer - (TmC - 0.889), ".000")

If Check1.Value = 1 Then InterChange
   Do While Timer - TmC < 0.32
   Dummy = DoEvents()
Loop
If Val(text3.Text) = 1 Then cTime.Caption = "CTime (S) = " & Format$(Timer - (TmC + 0.111), ".000")

If Check3.Value = 1 Then InterChg3
   Do While Timer - TmC < 0.43
   Dummy = DoEvents()
Loop
If Val(text3.Text) = 3 Then cTime.Caption = "CTime (S) = " & Format$(Timer - (TmC - 0.667), ".000")

If Check1.Value = 1 Then InterChange
   Do While Timer - TmC < 0.54
   Dummy = DoEvents()
Loop
If Val(text3.Text) = 1 Then cTime.Caption = "CTime (S) = " & Format$(Timer - (TmC + 0.333), ".000")

If Check4.Value = 1 Then InterChg4
   Do While Timer - TmC < 0.65
   Dummy = DoEvents()
Loop
If Val(text3.Text) = 4 Then cTime.Caption = "CTime (S) = " & Format$(Timer - (TmC - 0.444), ".000")

If Check1.Value = 1 Then InterChange
   Do While Timer - TmC < 0.76
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        Dummy = DoEvents()
      Loop
      If Val(text3.Text) = 1 Then cTime.Caption = "CTime (S) = " & Format$(Timer - (TmC + 0.556), ".000")

      If Check5.Value = 1 Then InterChg5
        Do While Timer - TmC < 0.87
        Dummy = DoEvents()
      Loop
      If Val(text3.Text) = 5 Then cTime.Caption = "CTime (S) = " & Format$(Timer - (TmC - 0.222), ".000")

      If Check1.Value = 1 Then InterChange
        Do While Timer - TmC < 0.98
        Dummy = DoEvents()
      Loop
      If Val(text3.Text) = 1 Then cTime.Caption = "CTime (S) = " & Format$(Timer - (TmC + 0.777), ".000")

      If Check6.Value = 1 Then InterChg6
        Do While Timer - TmC < 1.1
        Dummy = DoEvents()
      Loop
      If Val(text3.Text) = 6 Then cTime.Caption = "CTime (S) = " & Format$(Timer - (TmC), ".000")


  Wend

  End Sub

  Sub InterChg5()
      MSComm2.DTREnable = True 'Set RS-232 to 485 Converter (Tx)

       IOstring$ = "!" & ST_No(5) & GCS(5) & "++++" & "$#"
       ChkSUM (False)

    'If MSComm2.PortOpen = True Then MSComm2.Output = IOstring$
    If SFlag = False Then Exit Sub
    MSComm2.Output = IOstring$

    'ANTOutput = Text00Bias * 2.55
    If Val(text3.Text) = 5 Then Text1.Text = IOstring$ 'Display To Controller (GITP)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
LpN = 0
Do 'Rx (RS-485)
  LpN = LpN + 1
  Dummy = DoEvents()
Loop Until (MSComm2.OutBufferCount = 0 And LpN > 50) 'delay
MSComm2.DTREnable = False 'Set RS-232 to 485 Converter (Rx)



' Wait for data to come back to the serial port.
LpN = 0
Do
  LpN = LpN + 1
  Dummy = DoEvents()
Loop Until (MSComm2.InBufferCount >= 10 Or LpN > 500)
      If (LpN > 500 And MSComm2.InBufferCount <> 10) Then
       If MSComm2.PortOpen = True Then IOstring$ = MSComm2.Input
       TxtStatus(5).Text = IOstring$
       If MSComm2.InBufferCount = 0 Then
        If cB(5) <> 10 Then cB(5) = 10 Else cB(5) = 12
        Beep
        TxtStatus(5).Text = "Error"
        TxtStatus(5).ForeColor = QBColor(cB(5))
       End If
      Else
       If TxtStatus(5) <> "OnL" Then
        TxtStatus(5) = "OnL"
        TxtStatus(5).ForeColor = QBColor(0)
       End If
      End If
If LpN > 500 Then Exit Sub 'time is up

' Read the "GITP" response data in the serial port.
IOstring$ = MSComm2.Input

ChkSUM (True) 'Check DATA IN
If Val(text3.Text) = 5 Then Text2.Text = IOstring$ 'Display From Controller (GITP)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    If DataErr(0) = 0 Then
      Text5.Text = Mid(IOstring$, 5, 2)
      'ANTInput = Val("&H" & Mid(IOstring$, 5, 2))
      'ANTOutput = 200 'Val("&H" & Mid(IOstring$, 7, 2))
    Else
      DataErr(0) = 0
      errcount.Caption = "DATAErr = " & DataErr(1)
      Beep
      Exit Sub
    End If


If Check8.Value = 1 Then Text6.Text = Text5.Text
End Sub

Sub InterChg6()
 MSComm2.DTREnable = True 'Set RS-232 to 485 Converter (Tx)

     'IOstring$ = "!" & ST_No(6) & GCS(6) & "++++" & "$#"
     'ChkSUM (False)
       IOstring$ = "!" & ST_No(6) & GCS(6) & Text6.Text & "++" & "$#"
     'Vo = Val(Text00MV.Text)
     If Mid(IOstring$, 10, 1) <> "#" Then
       IOstring$ = "!" & ST_No(6) & GCS(6) & "0" & Text6.Text & "++" & "$#"
     Else
     End If
       ChkSUM (False)

  'If MSComm2.PortOpen = True Then MSComm2.Output = IOstring$
  If SFlag = False Then Exit Sub
  MSComm2.Output = IOstring$

  'ANTOutput = Text00Bias * 2.55
  If Val(text3.Text) = 6 Then Text1.Text = IOstring$ 'Display To Controller (GITP)

  LpN = 0
  Do  'Rx (RS-485)
    LpN = LpN + 1
    Dummy = DoEvents()
  Loop Until (MSComm2.OutBufferCount = 0 And LpN > 50) 'delay
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
MSComm2.DTREnable = False 'Set RS-232 to 485 Converter (Rx)



' Wait for data to come back to the serial port.
LpN = 0
Do
  LpN = LpN + 1
  Dummy = DoEvents()
Loop Until (MSComm2.InBufferCount >= 10 Or LpN > 500)
      If (LpN > 500 And MSComm2.InBufferCount <> 10) Then
       If MSComm2.PortOpen = True Then IOstring$ = MSComm2.Input
       TxtStatus(6).Text = IOstring$
       If MSComm2.InBufferCount = 0 Then
        If cB(6) <> 10 Then cB(6) = 10 Else cB(6) = 12
        Beep
        TxtStatus(6).Text = "Error"
        TxtStatus(6).ForeColor = QBColor(cB(6))
       End If
      Else
       If TxtStatus(6) <> "OnL" Then
        TxtStatus(6) = "OnL"
        TxtStatus(6).ForeColor = QBColor(0)
       End If
      End If
If LpN > 500 Then Exit Sub 'time is up

' Read the "GITP" response data in the serial port.
IOstring$ = MSComm2.Input

ChkSUM (True) 'Check DATA IN
If Val(text3.Text) = 6 Then Text2.Text = IOstring$ 'Display From Controller (GITP)

If DataErr(0) = 0 Then
  'ANTInput = Val("&H" & Mid(IOstring$, 5, 2))
  'ANTOutput = 200 'Val("&H" & Mid(IOstring$, 7, 2))
Else
  DataErr(0) = 0
  errcount.Caption = "DATAErr = " & DataErr(1)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        Beep
        Exit Sub
    End If

End Sub


Sub MVg_Visible(Vmv)
PicMV_Graph.Visible = Vmv
Label34(1).Visible = Vmv
Label12.Visible = Vmv
Label13.Visible = Vmv
Label15.Visible = Vmv
Label7.Visible = Vmv
Label9.Visible = Vmv
Label8.Visible = Vmv
Label10.Visible = Vmv
Line3.Visible = Vmv
Line4.Visible = Vmv
Line10.Visible = Vmv
Line8.Visible = Vmv
Line9.Visible = Vmv
GaugeMV.Visible = Vmv
End Sub

Sub PVg_Visible(Vpv As Boolean)
PicPV_Graph.Visible = Vpv
Label34(0).Visible = Vpv
Label14.Visible = Vpv
Label16.Visible = Vpv
Label17.Visible = Vpv
Label6.Visible = Vpv
Label3.Visible = Vpv
Label5.Visible = Vpv
Label4.Visible = Vpv
Line1.Visible = Vpv
Line2.Visible = Vpv
Line5.Visible = Vpv
Line6.Visible = Vpv
Line7.Visible = Vpv
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
GaugePV.Visible = Vpv
End Sub
Sub InterChg4()
    MSComm2.DTREnable = True 'Set RS-232 to 485 Converter (Tx)


     IOstring$ = "!" & ST_No(4) & GCS(4) & Hex(((Val(N04txt.Text) / 100) * 255) \ 1) & "++" & "$#"
     'Vo = Val(Text00MV.Text)
     If Mid(IOstring$, 10, 1) <> "#" Then
       IOstring$ = "!" & ST_No(4) & GCS(4) & "0" & Hex(((Val(N04txt.Text) / 100) * 255) \ 1) & "++" & "$#"
     Else
     End If


     'IOstring$ = "!" & ST_No(4) & GCS(4) & "++++" & "$#"
     ChkSUM (False)

  'If MSComm2.PortOpen = True Then MSComm2.Output = IOstring$
  If SFlag = False Then Exit Sub
  MSComm2.Output = IOstring$

  'ANTOutput = Text00Bias * 2.55
  If Val(text3.Text) = 4 Then Text1.Text = IOstring$ 'Display To Controller (GITP)

  LpN = 0
  Do  'Rx (RS-485)
     LpN = LpN + 1
     Dummy = DoEvents()
  Loop Until (MSComm2.OutBufferCount = 0 And LpN > 50) 'delay
  MSComm2.DTREnable = False 'Set RS-232 to 485 Converter (Rx)



' Wait for data to come back to the serial port.
LpN = 0
Do
   LpN = LpN + 1
   Dummy = DoEvents()
Loop Until (MSComm2.InBufferCount >= 10 Or LpN > 500)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            If (LpN > 500 And MSComm2.InBufferCount <> 10) Then
              If MSComm2.PortOpen = True Then IOstring$ = MSComm2.Input
              TxtStatus(4).Text = IOstring$
              If MSComm2.InBufferCount = 0 Then
                If cB(4) <> 10 Then cB(4) = 10 Else cB(4) = 12
                Beep
                TxtStatus(4).Text = "Error"
                TxtStatus(4).ForeColor = QBColor(cB(4))
              End If
            Else
              If TxtStatus(4) <> "OnL" Then
                TxtStatus(4) = "OnL"
                TxtStatus(4).ForeColor = QBColor(0)
              End If
            End If
          If LpN > 500 Then Exit Sub 'time is up

          ' Read the "GITP" response data in the serial port.
          IOstring$ = MSComm2.Input

          ChkSUM (True) 'Check DATA IN
          If Val(text3.Text) = 4 Then Text2.Text = IOstring$ 'Display From Controller (GITP)

          If DataErr(0) = 0 Then
            ANTOutput = Val("&H" & Mid(IOstring$, 5, 2))
            ANTInput = ANTOutput
            'ANTInput = ANTInputH
          Else
            DataErr(0) = 0
            errcount.Caption = "DATAErr = " & DataErr(1)
            Beep
            Exit Sub
          End If
        ' Alarm
        'If ANTInput / 2.55 > Val(Text00HighAlarm.Text) Or ANTInput / 2.55 < Val(Text00LowAlarm.Text) Then Beep


        'Show value (Gauge and Label)
        'Plot Graph
        If ((Val(text3.Text) = 3 Or Val(text3.Text) = 4) And Check7.Value = 1) Or _
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
(Val(text3.Text) = 4 And Check7.Value = 0) Then

If RfshG = True Then RefreshG
PlotG
RfshG = False
End If



End Sub

Sub IniVC()
"Initializing"
xxx = 1
ST_No = Array("", "01", "02", "03", "04", "05", "06")
GCS = Array("", "B", "M", "W", "W", "W", "W")
DataErr(1) = 0
errcount.Caption = "DATAErr = " & DataErr(1)
RfshG = False
text3_Change
"Initializing"
End Sub



Sub InterChg2()

  MSComm2.DTREnable = True 'Set RS-232 to 485 Converter (Tx)

   IOstring$ = "!" & ST_No(2) & GCS(2) & Hex(((Val(Text00SetPoint.Text) / 100) * 255) \ 1) & "++" & "$#"
   'Vo = Val(Text00MV.Text)
   If Mid(IOstring$, 10, 1) <> "#" Then
     IOstring$ = "!" & ST_No(2) & GCS(2) & "0" & Hex(((Val(Text00SetPoint.Text) / 100) * 255) \ 1) & "++" & "$#"
   Else
   End If
   ChkSUM (False)

'If MSComm2.PortOpen = True Then MSComm2.Output = IOstring$
If SFlag = False Then Exit Sub
MSComm2.Output = IOstring$
'ANTOutput = Text00Bias * 2.55
If Val(text3.Text) = 2 Then Text1.Text = IOstring$ 'Display To Controller (GITP)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
LpN = 0
Do 'Rx (RS-485)
  LpN = LpN + 1
  Dummy = DoEvents()
Loop Until (MSComm2.OutBufferCount = 0 And LpN > 50) 'delay
MSComm2.DTREnable = False 'Set RS-232 to 485 Converter (Rx)



' Wait for data to come back to the serial port.
LpN = 0
Do
  LpN = LpN + 1
  Dummy = DoEvents()
Loop Until (MSComm2.InBufferCount >= 10 Or LpN > 500)
      If (LpN > 500 And MSComm2.InBufferCount <> 10) Then
        If MSComm2.PortOpen = True Then IOstring$ = MSComm2.Input
        TxtStatus(2).Text = IOstring$
        If MSComm2.InBufferCount = 0 Then
          If cB(2) <> 10 Then cB(2) = 10 Else cB(2) = 12
          Beep
          TxtStatus(2).Text = "Error"
          TxtStatus(2).ForeColor = QBColor(cB(2))
        End If
      Else
        If TxtStatus(2) <> "OnL" Then
          TxtStatus(2) = "OnL"
          TxtStatus(2).ForeColor = QBColor(0)
        End If
      End If
If LpN > 500 Then Exit Sub 'time is up

' Read the "GITP" response data in the serial port.
IOstring$ = MSComm2.Input

ChkSUM (True) 'Check DATA IN
If Val(text3.Text) = 2 Then Text2.Text = IOstring$ 'Display From Controller (GITP)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    If DataErr(0) = 0 Then
      ANTInput = Val("&H" & Mid(IOstring$, 5, 2))
      ANTOutput = Val("&H" & Mid(IOstring$, 7, 2))
    Else
      DataErr(0) = 0
      errcount.Caption = "DATAErr = " & DataErr(1)
      Beep
      Exit Sub
    End If
  ' Alarm
  'If ANTInput / 2.55 > Val(Text00HighAlarm.Text) Or ANTInput / 2.55 < Val(Text00LowAlarm.Text) Then Beep


  'Show value (Gauge and Label)
  'Plot Graph
  If Val(text3.Text) = 2 Then
  If RfshG = True Then RefreshG
  PlotG
  RfshG = False
  End If
  End Sub

  Sub InterChg3()
     MSComm2.DTREnable = True 'Set RS-232 to 485 Converter (Tx)

      IOstring$ = "!" & ST_No(3) & GCS(3) & "++++" & "$#"
      ChkSUM (False)

    'If MSComm2.PortOpen = True Then MSComm2.Output = IOstring$
    If SFlag = False Then Exit Sub
    MSComm2.Output = IOstring$

    'ANTOutput = Text00Bias * 2.55
    If Val(text3.Text) = 3 Then Text1.Text = IOstring$ 'Display To Controller (GITP)

    LpN = 0
    Do  'Rx (RS-485)
      LpN = LpN + 1
      Dummy = DoEvents()
    Loop Until (MSComm2.OutBufferCount = 0 And LpN > 50) 'delay
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
MSComm2.DTREnable = False 'Set RS-232 to 485 Converter (Rx)



' Wait for data to come back to the serial port.
LpN = 0
Do
   LpN = LpN + 1
   Dummy = DoEvents()
Loop Until (MSComm2.InBufferCount >= 10 Or LpN > 500)
        If (LpN > 500 And MSComm2.InBufferCount <> 10) Then
         If MSComm2.PortOpen = True Then IOstring$ = MSComm2.Input
         TxtStatus(3).Text = IOstring$
         If MSComm2.InBufferCount = 0 Then
           If cB(3) <> 10 Then cB(3) = 10 Else cB(3) = 12
           Beep
           TxtStatus(3).Text = "Error"
           TxtStatus(3).ForeColor = QBColor(cB(3))
         End If
       Else
         If TxtStatus(3) <> "OnL" Then
           TxtStatus(3) = "OnL"
           TxtStatus(3).ForeColor = QBColor(0)
         End If
       End If
If LpN > 500 Then Exit Sub 'time is up

' Read the "GITP" response data in the serial port.
IOstring$ = MSComm2.Input

ChkSUM (True) 'Check DATA IN
If Val(text3.Text) = 3 Then Text2.Text = IOstring$ 'Display From Controller (GITP)

If DataErr(0) = 0 Then
   ANTInput = Val("&H" & Mid(IOstring$, 5, 2))
   'ANTOutput = 200 'Val("&H" & Mid(IOstring$, 7, 2))
Else
   DataErr(0) = 0
   errcount.Caption = "DATAErr = " & DataErr(1)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        Ypos1 = ANTInput - 255      ' Get vertical position.
        Ypos2 = ANTOutput - 255
        PicPV_Graph.Line (LXpos1, LYpos1)-(Xpos1, Ypos1), 44000 ' Draw GraphPV.
        PicMV_Graph.Line (LXpos2, LYpos2)-(Xpos2, Ypos2), 44000
        LXpos1 = Xpos1
        LXpos2 = Xpos2
        LYpos1 = Ypos1
        LYpos2 = Ypos2
        Line1st = False  'Clear graph begin status
End Sub

Sub RefreshG()
            If SFlag = True Then
IncX = 0
        'Clear both graph
        PicPV_Graph.Cls
        PicMV_Graph.Cls
        'Draw red line and red "SP"
        PicPV_Graph.Line (0, Val(Text00SetPoint.Text) * 2.55 - 255)-(3000, Val(Text00SetPoint.Text) * 2.55 - 255), QBColor(4)
        PicPV_Graph.CurrentX = 2700
          If Val(Text00SetPoint.Text) <= 75 Then
            PicPV_Graph.CurrentY = Val(Text00SetPoint.Text) * 2.55 - 190
          Else
            PicPV_Graph.CurrentY = Val(Text00SetPoint.Text) * 2.55 - 255
          End If
        PicPV_Graph.Print "SP"
        'Draw red line and red "BIAS"
        PicMV_Graph.Line (0, Val(Text00Bias.Text) * 2.55 - 255)-(3000, Val(Text00Bias.Text) * 2.55 - 255), QBColor(4)
        PicMV_Graph.CurrentX = 2500
          If Val(Text00Bias.Text) <= 75 Then
            PicMV_Graph.CurrentY = Val(Text00Bias.Text) * 2.55 - 190
          Else
            PicMV_Graph.CurrentY = Val(Text00Bias.Text) * 2.55 - 255
          End If
        PicMV_Graph.Print "BIAS"
        LXpos1 = IncX
        LXpos2 = IncX
        LYpos1 = ANTInput - 255
        LYpos2 = ANTOutput - 255
            End If
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
End Sub
Sub SelectText(txt As TextBox)
txt.SelStart = 0
txt.SelLength = Len(txt.Text)
End Sub

Sub ZoomMV_G()
        If SFlag = True Then
With PicMV_Graph
If .Height = 1000 Then


  .Top = 1350
  .Left = 1800
  .Height = 2000
  .Width = 6000

  .ScaleMode = 0 ' Set ScaleMode to
  .DrawWidth = 1  ' Set DrawWidth.
  .ScaleHeight = -258
  .ScaleWidth = 3000
  RefreshG
Else
  .Width = 3000
  .Height = 1000
  .Left = 4680
  .Top = 1800

  .ScaleMode = 0 ' Set ScaleMode to
  .DrawWidth = 1  ' Set DrawWidth.
  .ScaleHeight = -258
  .ScaleWidth = 3000
  RefreshG
End If
End With
        End If
End Sub

Sub ZoomPV_G()
        If SFlag = True Then
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
With PicPV_Graph
If PicPV_Graph.Height = 1000 Then


  .Top = 120
  .Left = 1800
  .Height = 2000
  .Width = 6000

  .ScaleMode = 0 ' Set ScaleMode to
  .DrawWidth = 1  ' Set DrawWidth.
  .ScaleHeight = -258
  .ScaleWidth = 3000
  RefreshG
Else
  .Width = 3000
  .Height = 1000
  .Left = 4680
  .Top = 360


  .Refresh
  .ScaleMode = 0 ' Set ScaleMode to
  .DrawWidth = 1  ' Set DrawWidth.
  .ScaleHeight = -258
  .ScaleWidth = 3000
  RefreshG
End If
End With
        End If
End Sub

Private Sub Check7_Click()
RfshG = True
Text4.Text = GCS(Val(text3.Text))

  If Val(text3.Text) = 4 Then
    If Check7.Value = 0 Then
    PVg_Visible (False)
    Else
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        PVg_Visible (True)
        End If
      End If

      If Val(text3.Text) = 3 Then
        If Check7.Value = 0 Then
        MVg_Visible (False)
        Else
        MVg_Visible (True)
        End If
      End If
    End Sub

    Private Sub CommandStart_Click()

     If SFlag = False Then
      If MSComm2.PortOpen = True Then MSComm2.PortOpen = False
       ' Use COM2.
       MSComm2.CommPort = 2
       ' 9600 baud, no parity, 8 data, and 1 stop bit.
       MSComm2.Settings = "9600,N,8,1"
       ' Handshacking RTS
       'MSComm2.Handshaking = 2
       ' Tell the control to read entire buffer when Input is used.
       MSComm2.InputLen = 0
       'MSComm2.RTSEnable = True
       MSComm2.DTREnable = True
       ' Open the port.
       MSComm2.PortOpen = True
       SFlag = True
       CommandStart.Caption = "STOP"
       StatusBar1.SimpleText = "Processing"



     PicPV_Graph.ScaleMode = 0 ' Set ScaleMode to
     PicPV_Graph.DrawWidth = 1  ' Set DrawWidth.
     PicPV_Graph.ScaleHeight = -258
     PicPV_Graph.ScaleWidth = 3000
     PicMV_Graph.ScaleMode = 0 ' Set ScaleMode to
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    PicMV_Graph.DrawWidth = 1  ' Set DrawWidth.
    PicMV_Graph.ScaleHeight = -258
    PicMV_Graph.ScaleWidth = 3000
    Line1st = True
    Slider00SetPoint.Value = Val(Text00SetPoint.Text)
    If SSOpt00Auto.Value = True Then
      Text00MV.Enabled = False
      Else
      Text00MV.Enabled = True
    End If
  Else

    MSComm2.PortOpen = False ' Close the serial port.
    StatusBar1.SimpleText = "Ready"
    CommandStart.Caption = "START"
    SFlag = False
  End If

DataErr(1) = 0
errcount.Caption = "DATAErr = " & DataErr(1)
CommuShare
End Sub

Sub InterChange()
If Line1st = True Then


    MSComm2.DTREnable = True 'Set RS-232 to 485 Converter (Tx)

    IOstring$ = "!" & ST_No(1) & GCS(1) & Hex(((Val(Text00Bias.Text) / 100) * 255) \ 1) & "++" & "$#"
    Vo = Val(Text00MV.Text)
    If Mid(IOstring$, 10, 1) <> "#" Then
      IOstring$ = "!" & ST_No(1) & GCS(1) & "0" & Hex(((Val(Text00Bias.Text) / 100) * 255) \ 1) & "++" & "$#"
    Else
    End If
    ChkSUM (False)

  'If MSComm2.PortOpen = True Then MSComm2.Output = IOstring$
  If SFlag = False Then Exit Sub
  MSComm2.Output = IOstring$
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    ANTOutput = Text00Bias * 2.55
    If Val(text3.Text) = 1 Then Text1.Text = IOstring$ 'Display To Controller (GITP)

    LpN = 0
    Do  'Rx (RS-485)
      LpN = LpN + 1
      Dummy = DoEvents()
    Loop Until (MSComm2.OutBufferCount = 0 And LpN > 50) 'delay
    MSComm2.DTREnable = False 'Set RS-232 to 485 Converter (Rx)
Else
      If Val(text3.Text) = 1 Then Text1.Text = LastOut$ 'Display To Controller (GITP)
      MSComm2.DTREnable = True
        'If MSComm2.PortOpen = True Then MSComm2.Output = LastOut$
        If SFlag = False Then Exit Sub
        MSComm2.Output = LastOut$
        LpN = 0
        Do  'Rx (RS-485)
          LpN = LpN + 1
          Dummy = DoEvents()
        Loop Until (MSComm2.OutBufferCount = 0 And LpN > 50)

      MSComm2.DTREnable = False
End If

  ' Wait for data to come back to the serial port.
  LpN = 0
  Do
    LpN = LpN + 1
    Dummy = DoEvents()
  Loop Until (MSComm2.InBufferCount >= 10 Or LpN > 500)
        If (LpN > 500 And MSComm2.InBufferCount <> 10) Then
          If MSComm2.PortOpen = True Then IOstring$ = MSComm2.Input
          TxtStatus(1).Text = IOstring$
          If MSComm2.InBufferCount = 0 Then
            If cB(1) <> 10 Then cB(1) = 10 Else cB(1) = 12
            Beep
            TxtStatus(1).Text = "Error"
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            TxtStatus(1).ForeColor = QBColor(cB(1))
          End If
        Else
          If TxtStatus(1) <> "OnL" Then
            TxtStatus(1) = "OnL"
            TxtStatus(1).ForeColor = QBColor(0)
          End If
        End If
    If LpN > 500 Then Exit Sub 'time is up

    ' Read the "GITP" response data in the serial port.
    IOstring$ = MSComm2.Input

    ChkSUM (True) 'Check DATA IN
    If Val(text3.Text) = 1 Then Text2.Text = IOstring$ 'Display From Controller (GITP)
    If DataErr(0) = 0 Then
      ANTInput = Val("&H" & Mid(IOstring$, 5, 2))
    Else
      If Line1st = True Then ANTInput = Text00SetPoint * 2.55
      DataErr(0) = 0
      errcount.Caption = "DATAErr = " & DataErr(1)
      Beep
    End If
  'PV Alarm
  If ANTInput / 2.55 > Val(Text00HighAlarm.Text) Or ANTInput / 2.55 < Val(Text00LowAlarm.Text) Then Beep

  'dim ,get variable and calculate PID function
  Dim en__1, SP, PV, Kp, Ki, T, Kd As Double

  If Line1st = True Then Vo = Val(Text00Bias.Text)
  Kp = Val(Text00Kp.Text) '%/%
  Ki = Val(Text00Ki.Text)  '%/%
  Kd = Val(Text00Kd.Text) '%/%
  SP = Val(Text00SetPoint.Text) '%
  PV = (ANTInput / 255) * 100 '%
        '  If Line1st = True Then
        '    PV = SP
        '  Else
        '    PV = (ANTInput / 255) * 100 '%
        '  End If
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
en_2 = CDbl(en_1)
en_1 = CDbl(en)
en = SP - PV '%
T = 0.222 'Sec
'en__1 = 2 * en - en_1
Von_1 = CDbl(Vo)


  Vo = Von_1 + Kp * (en - en_1) + Ki * en * T + (Kd / T) * (en - 2 * en_1 + en_2)

    If Vo > 100 Then Vo = 100   'Over 100% clip
    If Vo < 0 Then Vo = 0       'Lower  0% clip

'Check and do (auto/manual)
'If SSOpt00Auto.Value = True Then Text00MV.Text = Format(((ANTOutput / 255) * 100), "0.00")

  If SSOpt00Manual.Value = True Then
    IOstring$ = "!" & ST_No(1) & GCS(1) & Hex(((Val(Text00MV.Text) / 100) * 255) \ 1) & "++" & "$#"
    Vo = Val(Text00MV.Text)
    If Mid(IOstring$, 10, 1) <> "#" Then
      IOstring$ = "!" & ST_No(1) & GCS(1) & "0" & Hex(((Val(Text00MV.Text) / 100) * 255) \ 1) & "++" & "$#"
    Else
    End If
  Else


  'If SSChk00Rev.Value = False Then 'Direct/Revert Acting
  If (SSChk00Rev.Value Xor NOv.Value) = False Then 'Direct/Revert Acting

    IOstring$ = "!" & ST_No(1) & GCS(1) & Hex((((100 - Vo) / 100) * 255) \ 1) & "++" & "$#"
    If Mid(IOstring$, 10, 1) <> "#" Then IOstring$ = "!" & ST_No(1) & GCS(1) & "0" & Hex((((100 - Vo) / 100) * 255) \ 1) &
"++" & "$#"

  Else

    IOstring$ = "!" & ST_No(1) & GCS(1) & Hex(((Vo / 100) * 255) \ 1) & "++" & "$#"
    If Mid(IOstring$, 10, 1) <> "#" Then IOstring$ = "!" & ST_No(1) & GCS(1) & "0" & Hex(((Vo / 100) * 255) \ 1) & "++" &
"$#"

  End If
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    End If

    ANTOutput = Val("&H" & Mid(IOstring$, 5, 2))
    If SSOpt00Auto.Value = True Then Text00MV.Text = Format(((ANTOutput / 255) * 100), "0.00")  'BUMP LESS TRANSFER

    ChkSUM (False) ' Create ChkSum Byte for DATA OUT
    LastOut$ = IOstring$

    'Show value (Gauge and Label)
    'Plot Graph
    'If RfshG = True Then RefreshG
    If Val(text3.Text) = 1 Then
    If RfshG = True Then RefreshG
    PlotG
    RfshG = False
    End If
    End Sub

    Sub ChkSUM(INflag As Boolean)
    Dim SumNotC As Integer
    Dim V_Block(9) As String

    V_Block(0) = Mid(IOstring$, 1, 1)
    V_Block(1) = Mid(IOstring$, 2, 1)
    V_Block(2) = Mid(IOstring$, 3, 1)
    V_Block(3) = Mid(IOstring$, 4, 1)
    V_Block(4) = Mid(IOstring$, 5, 1)
    V_Block(5) = Mid(IOstring$, 6, 1)
    V_Block(6) = Mid(IOstring$, 7, 1)
    V_Block(7) = Mid(IOstring$, 8, 1)
    V_Block(8) = Mid(IOstring$, 9, 1)
    V_Block(9) = Mid(IOstring$, 10, 1)

    SumNotC = Asc(Mid(IOstring$, 2, 1)) + _
    Asc(Mid(IOstring$, 3, 1)) + _
    Asc(Mid(IOstring$, 4, 1)) + _
    Asc(Mid(IOstring$, 5, 1)) + _
    Asc(Mid(IOstring$, 6, 1)) + _
    Asc(Mid(IOstring$, 7, 1)) + _
    Asc(Mid(IOstring$, 8, 1))
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
SumNotC = SumNotC - (256 * (SumNotC \ 256))
ChkS = Chr(SumNotC)

If INflag = False Then
   IOstring$ = V_Block(0) & V_Block(1) & V_Block(2) _
   & V_Block(3) & V_Block(4) & V_Block(5) & V_Block(6) _
   & V_Block(7) & ChkS & V_Block(9)
Else
   If StrComp(V_Block(8), ChkS, 0) = 0 Then
      DataErr(0) = 0
   Else
      DataErr(0) = 1
      DataErr(1) = DataErr(1) + 1
   End If
   If V_Block(3) = "E" Then
   DataErr(1) = DataErr(1) + 1
   errcount.Caption = "DATAErr = " & DataErr(1)
   Beep
   End If
End If
End Sub



Private Sub Form_Load()
SFlag = False
IniVC
End Sub



Private Sub Form_Unload(Cancel As Integer)
Unload Me




End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Private Sub N04txt_GotFocus()
Call SelectText(N04txt)
End Sub

Private Sub N04txt_KeyUp(KeyCode As Integer, Shift As Integer)
If Val(N04txt.Text) = 0 And (N04txt.Text <> "0" And N04txt.Text <> "") Then N04txt.Text = ""
   If Val(N04txt.Text) < 0 Then N04txt.Text = "0"
   If Val(N04txt.Text) > 100 Then N04txt.Text = "100"
End Sub


Private Sub PicMV_Graph_Click()
If PicPV_Graph.Height = 2000 Then ZoomPV_G
ZoomMV_G
End Sub

Private Sub PicPV_Graph_Click()
If PicMV_Graph.Height = 2000 Then ZoomMV_G
ZoomPV_G
End Sub

Private Sub Slider00Setpoint_Scroll()
Text00SetPoint.Text = Slider00SetPoint.Value
End Sub



Private Sub SSOpt00Manual_Click(Value As Integer)
   If SSOpt00Auto.Value = True Then
      Text00MV.Enabled = False
      Else
      Text00MV.Enabled = True
   End If
End Sub

Private Sub SSOpt00Auto_Click(Value As Integer)
   If SSOpt00Auto.Value = True Then
      Text00MV.Enabled = False
      Else
      Text00MV.Enabled = True
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    End If
End Sub


Private Sub Text00Bias_GotFocus()
Call SelectText(Text00Bias)
End Sub

Private Sub Text00HighAlarm_GotFocus()
Call SelectText(Text00HighAlarm)
End Sub


Private Sub Text00Kd_GotFocus()
Call SelectText(Text00Kd)
End Sub


Private Sub Text00Ki_GotFocus()
Call SelectText(Text00Ki)
End Sub


Private Sub Text00Kp_GotFocus()
Call SelectText(Text00Kp)
End Sub


Private Sub Text00LowAlarm_GotFocus()
Call SelectText(Text00LowAlarm)
End Sub


Private Sub Text00MV_GotFocus()
Call SelectText(Text00MV)
End Sub

Private Sub Text00MV_KeyUp(KeyCode As Integer, Shift As Integer)
If Val(Text00MV.Text) = 0 And (Text00MV.Text <> "0" And Text00MV.Text <> "") Then Text00MV.Text = ""
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    If Val(Text00MV.Text) < 0 Then Text00MV.Text = "0"
    If Val(Text00MV.Text) > 100 Then Text00MV.Text = "100"
'Call SelectText(Text00MV)
End Sub


Private Sub Text00SetPoint_Change()
If Val(Text00SetPoint.Text) = 0 And (Text00SetPoint.Text <> "0" And Text00SetPoint.Text <> "") Then Text00SetPoint.Text =
""
    If Val(Text00SetPoint.Text) < 0 Then Text00SetPoint.Text = "0"
    If Val(Text00SetPoint.Text) > 100 Then Text00SetPoint.Text = "100"
    Slider00SetPoint.Value = Val(Text00SetPoint.Text)
    RefreshG
End Sub


Private Sub Text00Bias_Change()
If Val(Text00Bias.Text) = 0 And (Text00Bias.Text <> "0" And Text00Bias.Text <> "") Then Text00Bias.Text =""
    If Val(Text00Bias.Text) < 0 Then Text00Bias.Text = "0"
    If Val(Text00Bias.Text) > 100 Then Text00Bias.Text = "100"
RefreshG
End Sub


Private Sub Text00SetPoint_GotFocus()
Call SelectText(Text00SetPoint)
End Sub

Private Sub text3_Change()
If Val(text3.Text) > 6 Or Val(text3.Text) < 1 Then text3.Text = "01"
VScroll1.Value = Val(text3.Text)
RfshG = True
Text4.Text = GCS(Val(text3.Text))


If (Val(text3.Text) = 5 Or Val(text3.Text) = 6) Then
    PVg_Visible (False)
    MVg_Visible (False)
    SliderTimeX.Visible = False
    Label33.Visible = False
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            Label35.Visible = False
            Label36.Visible = False
            Label37.Visible = False
            N04txt.Visible = False
            Label41.Visible = True
            Label40.Visible = True
            Label38.Visible = True
            Label39.Visible = True
            Text5.Visible = True
            Text6.Visible = True
        Else

        If Val(text3.Text) = 4 Then
           If Check7.Value = 0 Then PVg_Visible (False)
        Else
           PVg_Visible (True)
        End If

        If Val(text3.Text) = 3 Then
           If Check7.Value = 0 Then MVg_Visible (False)
        Else
           MVg_Visible (True)
        End If


    SliderTimeX.Visible = True
    Label33.Visible = True
    Label35.Visible = True
    Label36.Visible = True
    Label37.Visible = True
    N04txt.Visible = True
        Label41.Visible = False
        Label40.Visible = False
        Label38.Visible = False
        Label39.Visible = False
        Text5.Visible = False
        Text6.Visible = False
    End If
    End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Private Sub text3_GotFocus()
Call SelectText(text3)
End Sub


Private Sub text3_KeyUp(KeyCode As Integer, Shift As Integer)
Call SelectText(text3)
End Sub

Private Sub Text4_Change()
If Len(Text4.Text) <> 1 Then Text4.Text = "W"
GCS(Val(text3.Text)) = Text4.Text
End Sub


Private Sub Text4_GotFocus()
Call SelectText(Text4)
End Sub


Private Sub Text4_KeyUp(KeyCode As Integer, Shift As Integer)
Call SelectText(Text4)
End Sub


Private Sub VScroll1_Change()
text3.Text = Format$(Val(VScroll1), "00")
End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
; FILENAME    N01V5.ASM
; DESCRIPTION  PROGRAM FOR Small Scale DCS (Analog I/O Controller)
; HARDWARE    ANT-32 & DMC202
; ASSEMBLER   SXA51
; START-DATE  28/08/96
; SOFTWARE ENG  KMITL Power User
; COMPANY     KMITL , KMITNB

;********** SET VARIABLE **********
X1LCDWRC     EQU   0FA00H          ;LCD WRITE CONTROL (ANT-31PJ)
X1LCDRDC     EQU   0FA01H          ;LCD READ CONTROL
X1LCDWRD     EQU   0FA02H          ;LCD WRITE DATA
X1LCDRDD     EQU   0FA03H          ;LCD READ DATA

PORT1A  EQU  0F800H
PORT1B  EQU  0F801H
PORT1C  EQU  0F802H
PORT1P  EQU  0F803H       ;8255 CONTROL PORT1

PORT2A  EQU  0FC00H
PORT2B  EQU  0FC01H
PORT2C  EQU  0FC02H
PORT2P  EQU  0FC03H       ;8255 CONTROL PORT2

;********** Dim Variable **********

V0   EQU   30H   ;A Buffer
V1   EQU   31H   ;USED
V2   EQU   32H   ;USED
V3   EQU   33H   ;Percent X---.-
V4   EQU   34H   ;Percent -X-.-
V5   EQU   35H   ;Percent --X.-
V6   EQU   36H   ;Percent ---.X
V7   EQU   37H

V_BlockST    EQU   40H
V_Block00    EQU   41H
V_Block01    EQU   42H
V_Block02    EQU   43H
V_Block03    EQU   44H
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
V_Block04     EQU   45H
V_Block05     EQU   46H
V_Block06     EQU   47H
V_Block07     EQU   48H
V_BlockEND    EQU   49H

LL_BUFH       EQU   2AH
LL_BUFP       EQU   51H
LL_BUF        EQU   52H



;*****************************;
;    SERIAL USE INT
;    ANT-32,REM31
;*****************************;
     ORG   0000H
     CLR   EA    ;All Interrupt Disable
     AJMP  RES

     ORG   0023H  ;REM31 INT Vector RI+TI : MCS-51(0023H)
     JBC   RI,RECV                ;TI+RI Vector Action
     LJMP  OUT    ;if TI reti OUT

RECV:

   PUSH  02H   ;testTO
   PUSH  03H   ;testTO
   PUSH  04H   ;testTO

     LCALL  RxBlock  ;Save MV to V_Block
     mov    a,V_BlockST
     cjne   a,#21H ,OUT
     mov    a,V_Block00
     cjne   a,#30H ,OUT
     mov    a,V_Block01
     cjne   a,#31H ,OUT

     LCALL  MVO     ;Get MV from Block,Convert to HEX,OUT PORT2B,(LCD)

     lcall  send
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    JNB   TI,$
    CLR   TI
    SETB  ES

OUT:
  POP   04H  ;testTO
  POP   03H  ;testTO
  POP   02H  ;testTO
    RETI

SEND:
    setb  p3.4
    LCALL  PVIB   ;IN PV,Save to V_Block,(LCD)
    LCALL  CHKSUM
    MOV   V_Block07,A
    LCALL  TxBlock ;Tx V_Block
    clr   p3.4
    RET

    ORG   0100H
;***************** Reset *****************
RES:  MOV   R2,#40H       ;POWER UP DELAY
RES1:  MOV   R3,#0
    DJNZ  R3,$
    DJNZ  R2,RES1

    MOV   R2,#3FH
    MOV   R0,#20H
XX:   MOV   @R0,#0
    INC   R0
    DJNZ  R2,XX

;***************** Set Const ***************
ST:         LCALL  INIv
          clr   p3.4
;SetIntV:      mov   dptr,#8008h             ;Not use for EPROM
;          mov   a,#80h                ;
;          movx  @dptr,a               ;
;          inc   dptr                  ;
;          mov   a,#23h                ;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
;       movx  @dptr,a           ;

OVER:
    MOV   SCON,#50H             ;Serial Mode 1 9600
    ;MOV   PCON,#10000000B      ;x2 = 19200
    MOV   TMOD,#20H             ;
    MOV   TH1,#0FDH             ;
    SETB  TR1                   ;
    ;SETB  EA                   ;
    SETB  ES                    ;


        MOV   DPTR,#PORT1P      ;SET 8255 CONTROL PORT1
        MOV   A,#90H            ;
        MOVX  @DPTR,A           ;

        MOV   DPTR,#PORT2P      ;SET 8255 CONTROL PORT2
        MOV   A,#90H            ;
        MOVX  @DPTR,A           ;

            MOV   R2,#1
        LCALL  DELAY            ;Set LCD
        MOV   A,#00111000B ;FUNCTION SET   ;
        LCALL  X1LCDWI          ;
        MOV   A,#00001100B ;DISPLAY ON/OFF  ;
        LCALL  X1LCDWI          ;
        MOV   A,#01H    ;CLEAR      ;
        LCALL  X1LCDWI          ;
                ;

        MOV   DPTR,#MAINT2      ;LOAD & Display
        LCALL  X1LCDLDP         ;Form
        MOV   R2,#10H           ;
        LCALL  DELAY            ;

        MOV   DPTR,#MAINT4      ;LOAD & Display
        LCALL  X1LCDLDP         ;Form
        MOV   R2,#10H           ;
        LCALL  DELAY            ;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        MOV   DPTR,#MAINT3          ;LOAD & Display
        LCALL X1LCDLDP              ;Form
        MOV   R2,#15H               ;
        LCALL DELAY                 ;

        MOV   DPTR,#MAINT0          ;LOAD & Display
        LCALL X1LCDLDP              ;Form
        MOV   R2,#02H               ;
        LCALL DELAY                 ;

        SETB  EA                    ;All Interrupt Enable                    ;

Loop001: cpl  p1.7                  ;testTO
        MOV   R2,#01H               ;testTO
        LCALL DELAY                 ;testTO
        cpl   p1.6                  ;testTO
        MOV   R2,#01H               ;testTO
        LCALL DELAY                 ;testTO
        cpl   p1.5                  ;testTO
        MOV   R2,#01H               ;testTO
        LCALL DELAY                 ;testTO
        Ljmp  loop001               ;testTO

        SJMP  $ ;add1               ; STOP

INIv:
    MOV   V_BlockST,#00100001B  ;!
    MOV   V_BlockEND,#00100011B ;#

    MOV   V_Block00,#30H    ;0   ST.No.
    MOV   V_Block01,#31H    ;1   ST.No.
    MOV   V_Block02,#57H    ;W   General Command & Status
    MOV   V_Block03,#2BH    ;+   DATA H
    MOV   V_Block04,#2BH    ;+   DATA L
    MOV   V_Block05,#2BH    ;+   MultiPurpose
    MOV   V_Block06,#2BH    ;+   MultiPurpose
    MOV   V_Block07,#24H    ;$   ChkSUM

    RET
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
;MAIN1:


MAINT0: DB    "ST.No.01 PV = ---.-%"          ;LCD Form Table
    DB    " (PIDc) MV = ---.-%"          ;
MAINT1: DB    "----  PV --H [---]"          ;LCD Form Table
    DB    "----  MV --H [---]"          ;
MAINT2: DB    "  SmallScale DCS  "          ;
    DB    " - Initializing - "          ;
MAINT3: DB    " ST.No.01 ANALOG I/O"          ;
    DB    "  ####  CONTROLLER"          ;
MAINT4: DB    ">> SmallScale DCS <<"          ;
    DB    " - Initializing - "          ;

;************** TxBlock ************          ;Tx V_Block
                    ;
TxBlock:    MOV   R0,#40H          ;
        LCALL  mDELAY  ;TEST RS-485;
TxBLoop:    MOV   A,@R0          ;
        LCALL  Tx1Byte          ;
        INC   R0          ;
        CJNE   R0,#4AH,TxBLoop          ;



        RET          ;
                    ;
                    ;
; ********** Tx1Byte SUB **********          ;
; SEND 1 BYTE          ;
; IN = A          ;
; REG = NO          ;
                    ;
Tx1Byte:    CLR   TI          ;
        CLR   ES          ;
        MOV   SBUF,A          ;
        JNB   TI,$          ;
        RET          ;

;************** RxBlock ************
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
RxBlock:    MOV   dptr,#0000H    ;testTO

            MOV   A,SBUF
            ;JNB   RI,$          ;testTO;WAIT FOR RECEIVE OK
            mov  V_BlockST,A     ;SAVE 1ST from PC TO BLOCK
            cjne  a,#21H,RxOUT
          LCALL  Rx1Byte         ;ST.No. Byte 1
            mov  V_Block00,A     ;SAVE from PC TO BLOCK
            cjne  a,#30H,RxOUT
          LCALL  Rx1Byte         ;ST.No. Byte 2
            mov  V_Block01,A     ;SAVE from PC TO BLOCK
            cjne  a,#31H,RxOUT
          LCALL  Rx1Byte         ;GCS Byte

                CJNE   A,V_Block02,NotEq     ;Change form?
                CLR    PSW.1                 ;<> = 1
                SJMP   EqOut                 ;== 0
                NotEq:
                SETB   PSW.1
                EqOut:

            mov  V_Block02,A     ;SAVE from PC TO BLOCK
          LCALL  Rx1Byte
            mov  V_Block03,A     ;SAVE MV from PC TO BLOCK
          LCALL  Rx1Byte
            mov  V_Block04,A     ;SAVE MV from PC TO BLOCK
          LCALL  Rx1Byte      ;+
            mov  V_Block05,A     ;SAVE from PC TO BLOCK
          LCALL  Rx1Byte      ;+
            mov  V_Block06,A     ;SAVE from PC TO BLOCK
          LCALL  Rx1Byte      ;ChkSum
            mov  V_Block07,A     ;SAVE from PC TO BLOCK
          LCALL  Rx1Byte      ;END (#)
            mov  V_BlockEND,A    ;SAVE from PC TO BLOCK

          LCALL  FChange         ;Change Form

          MOV    A,V_Block02     ;If GCS <> B then goto RxOUT
          CJNE   A,#42H,RxOUT    ;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        LCALL  V_LCD

RxOUT:      LCALL  CHKSUM           ;ChkSum DATA IN
            CJNE   A,V_Block07,ChkErr   ;if Err GCS = E
            SJMP   ChkOK            ;
        ChkErr: MOV    V_Block02,#45H      ;(E)
        ChkOK:                      ;
            RET
; ********** Rx1Byte SUB **********
Rx1Byte:    INC    dptr             ;testTO
            mov    a,dph            ;testTO
            cjne   a,#045H,jovr     ;testTO
            mov    a,dpl            ;testTO
            cjne   a,#0ffH,jovr     ;testTO
            CLR    P1.0             ;testTO
         jovr:                      ;testTO
            mov    a,dph            ;testTO
            cjne   a,#055H,jovr2    ;testTO
            mov    a,dpl            ;testTO
            cjne   a,#0FFH,jovr2    ;testTO
            CLR    P1.1             ;testTO
         jovr2:                     ;testTO


            JNB    RI,Rx1byte       ;testTO;WAIT FOR RECEIVE OK
            CLR    RI

            ; ;MOV    A,#088H          ;testTO
            ; ;LCALL  X1LCDWI          ;testTO
            ; ;MOV    A,dph            ;testTO
            ; ;LCALL  X1LCDWD          ;testTO

            MOV    A,SBUF

            RET


;************* Form Change ****************
FChange:
            JNB    PSW.1,FChOut
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        MOV   A,V_Block02       ;If <> B then goto ifA
        CJNE  A,#42H,ifA   ;B
        MOV   DPTR,#MAINT1          ;LOAD & Display
        LCALL X1LCDLDP              ;Form
        ;MOV   R2,#20H              ;
        ;LCALL DELAY                ;
        SJMP  FChOut

ifA:    ;MOV   A,V_Block02      ;If <> A then goto FChOut
        ;CJNE  A,#41H,FChOut ;A
        MOV   DPTR,#MAINT0          ;LOAD & Display
        LCALL X1LCDLDP              ;Form
        ;MOV   R2,#20H              ;
        ;LCALL DELAY                ;

FChOut:     RET

;************** V_Block to LCD (from PC) **********
V_LCD:
        MOV   A,#080H
              LCALL X1LCDWI
        MOV   A,V_BlockST
              LCALL X1LCDWD

        MOV   A,#081H
              LCALL X1LCDWI
        MOV   A,V_Block00
              LCALL X1LCDWD

        MOV   A,#082H
              LCALL X1LCDWI
        MOV   A,V_Block01
              LCALL X1LCDWD

        MOV   A,#083H
              LCALL X1LCDWI
        MOV   A,V_Block02
              LCALL X1LCDWD

        MOV   A,#084H
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            LCALL  X1LCDWI
        MOV    A,V_Block03
            LCALL  X1LCDWD


        MOV    A,#0C0H
            LCALL  X1LCDWI
        MOV    A,V_Block04
            LCALL  X1LCDWD

        MOV    A,#0C1H
            LCALL  X1LCDWI
        MOV    A,V_Block05
            LCALL  X1LCDWD

        MOV    A,#0C2H
            LCALL  X1LCDWI
        MOV    A,V_Block06
            LCALL  X1LCDWD

        MOV    A,#0C3H
            LCALL  X1LCDWI
        MOV    A,V_Block07
            LCALL  X1LCDWD


        MOV    A,#0C4H
            LCALL  X1LCDWI
        MOV    A,V_BlockEND
            LCALL  X1LCDWD

        RET


;***********MV OUT SUB************

MVO:

        mov   r2,v_block03          ;MV from V_Block
        mov   r3,v_block04          ;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
lcall atoh  ;R2,R3 -> A        ;Convert to HEX
mov  v2,a                  ;Save to V2

;mov   p1,a
;MOV   A,#0FFH ;V2              ;OUT MV to Port1B
MOV   DPTR,#PORT1B             ;OUT MV to Port1B
MOVX  @DPTR,A                  ;

MOV   A,V_Block02              ;if GCS <> B then
CJNE  A,#42H,MVP               ;goto MVP

MOV   A,V2
LCALL  HTOA                    ;Display MV to LCD
MOV   A,#0CBH                  ;HEX
LCALL  X1LCDWI                   ;
MOV   A,R2                 ;
LCALL  X1LCDWD                   ;
MOV   A,#0CCH              ;
LCALL  X1LCDWI                   ;
MOV   A,R3                 ;
LCALL  X1LCDWD                   ;

MOV   A,V2                 ;Display MV to LCD
LCALL  BINBCD                    ;BCD
MOV   A,HUND               ;
LCALL  HTOA                ;
MOV   A,#0D0H              ;
LCALL  X1LCDWI                   ;
MOV   A,R3                 ;
LCALL  X1LCDWD                   ;
MOV   A,TENONE                   ;
LCALL  HTOA                ;
MOV   A,#0D1H              ;
LCALL  X1LCDWI                   ;
MOV   A,R2                 ;
LCALL  X1LCDWD                   ;
MOV   A,#0D2H              ;
LCALL  X1LCDWI                   ;
MOV   A,R3                 ;
LCALL  X1LCDWD                   ;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        SJMP   MVOout       ;goto MVOout

MVP:      MOV    V3,#0CEH              ;DISPLAY MV TO LCD
        MOV    V4,#0CFH              ; (%)
        MOV    V5,#0D0H              ;
        MOV    V6,#0D2H              ;
        MOV    LL_BUFH,V2             ;
        LCALL  Percent               ;

MVOout:     RET

;**********PV IN & to V_Block SUB**********

PVIB:

        mov    dptr,#PORT1A                ;Get input
        movx   a,@dptr                     ;
        MOV    V1,A   ;SAVE INPUT TO V1          ;Save to
                                  ;V_Block
        MOV    A,V1                        ;
        LCALL  HTOA                        ;
        mov  V_Block03,r2  ;SAVE PV TO V_BLOCK    ;
        mov  V_Block04,r3  ;SAVE PV TO V_BLOCK    ;

        MOV    A,V_Block02                 ;if GCS <> B then
        CJNE   A,#42H,PVP                  ;goto PVP

        MOV    A,V1
        MOV    A,#8BH                  ;DISPLAY PV TO LCD
        LCALL  X1LCDWI                 ;HEX
        MOV    A,R2                  ;
        LCALL  X1LCDWD                     ;
        MOV    A,#8CH                ;
        LCALL  X1LCDWI                 ;
        MOV    A,R3                  ;
        LCALL  X1LCDWD                     ;

        MOV    A,V1                  ;DISPLAY PV TO LCD
        LCALL  BINBCD                  ;BCD
        MOV    A,HUND                  ;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        LCALL  HTOA              ;
        MOV    A,#90H            ;
        LCALL  X1LCDWI           ;
        MOV    A,R3              ;
        LCALL  X1LCDWD           ;
        MOV    A,TENONE          ;
        LCALL  HTOA              ;
        MOV    A,#91H            ;
        LCALL  X1LCDWI           ;
        MOV    A,R2              ;
        LCALL  X1LCDWD           ;
        MOV    A,#92H            ;
        LCALL  X1LCDWI           ;
        MOV    A,R3              ;
        LCALL  X1LCDWD           ;
        SJMP   PVIout      ;goto PVIout

PVP:    MOV    V3,#08EH          ;DISPLAY PV TO LCD
        MOV    V4,#08FH          ; (%)
        MOV    V5,#090H          ;
        MOV    V6,#092H          ;
        MOV    LL_BUFH,V1        ;
        LCALL  Percent           ;

PVIout: RET



; ********** XLCDWI SUB ********** (ANT-32)
; LCD WRITE INSTRUCTION (RS=0)
; IN = A
; REG = A

X1LCDWI:    PUSH   DPH
            PUSH   DPL
            MOV    DPTR,#X1LCDWRC
            MOVX   @DPTR,A
            MOV    DPTR,#X1LCDRDC
X1LCDWI1:   MOVX   A,@DPTR          ;WAIT FOR BF=0
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                JB    ACC.7,X1LCDWI1
                POP   DPL
                POP   DPH
                RET

; ********** XLCDWD SUB ********** (ANT-32)
; LCD WRITE DATA (RS=1)
; IN = A
; REG = A

X1LCDWD:    PUSH  DPH
                PUSH  DPL
                MOV   DPTR,#X1LCDWRD
                MOVX  @DPTR,A
                MOV   DPTR,#X1LCDRDC
X1LCDWD1:   MOVX  A,@DPTR          ;WAIT FOR BF=0
                JB    ACC.7,X1LCDWD1
                POP   DPL
                POP   DPH
                RET

; ********** XLCDLDP SUB ********** (ANT-32)
; LOAD PMEM TO LCD-MODULE (DMC202)
; IN = DPTR START BLOCK
; REG = A,R2,DPTR

X1LCDLDP:   MOV   A,#80H           ;SET ADDRESS LINE 1
                LCALL X1CDLDPS
                MOV   A,#0C0H      ;SET ADDRESS LINE 2
                LCALL X1CDLDPS
                RET
X1CDLDPS:   LCALL X1LCDWI          ;LOAD ONE LINE
                MOV   R2,#20            ;20 CHAR.
X2CDLDPS:   CLR   A
                MOVC  A,@A+DPTR         ;PMEM=MOVC
                LCALL X1LCDWD           ;WRITE DATA
                INC   DPTR
                DJNZ  R2,X2CDLDPS
                RET
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
;******** HTOA SUB ***********
;CONVERT HEX TO ASCII
;IN = A
;OUT = R2,R3
;REG = A,R2,R3
HTOA:   PUSH  ACC
        SWAP  A
        LCALL HTOAS
        MOV   R2,A
        POP   ACC
        LCALL HTOAS
        MOV   R3,A
        RET
HTOAS:  ANL   A,#0FH
        CJNE  A,#0AH,$+3
        JNC   HTOAS1
        ORL   A,#30H
        RET
HTOAS1: SUBB  A,#9
        ORL   A,#40H
        RET


; ********** ATOH SUB **********
; ASCII TO HEX CONVERT
; IN = R2,R3  30H,41H
; OUT = A     0AH
; REG = A,R2

ATOH:  MOV   A,R2
    LCALL  ATOHS
    SWAP   A
    MOV    R2,A
    MOV    A,R3
    LCALL  ATOHS
    ORL    A,R2
    RET
ATOHS: CJNE  A,#'A',$+3
    JC     ATOHS1
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        ADD   A,#9
ATOHS1: ANL   A,#0FH
        RET

;********** BIN to BCD **********
HUND      EQU   21H
TENONE    EQU   22H

BINBCD:   MOV   B,#100      ;DIVIDE INPUT BY 100 TO
          DIV   AB
          MOV   HUND,A
          MOV   A,#10
          XCH   A,B
          DIV   AB
          SWAP  A
          ADD   A,B
          MOV   TENONE,A
          RET

; ********** CHKSUM SUB **********
;Sum V_Block00-06 without carry
;IN = V_Block00-06
;OUT = A

CHKSUM:
          MOV   A,#00H
          ADD   A,V_Block00
          ADD   A,V_Block01
          ADD   A,V_Block02
          ADD   A,V_Block03
          ADD   A,V_Block04
          ADD   A,V_Block05
          ADD   A,V_Block06

          RET
; ********** DELAY SUB **********
; DELAY SUBROUTINE
; IN  = R2
; REG = R2,R3,R4
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
DELAY:      MOV   R3,#0
DELAY1:     MOV   R4,#0
                  DJNZ  R4,$
                  DJNZ  R3,DELAY1
                  DJNZ  R2,DELAY
                  RET

; ********** mDELAY SUB **********
; mDELAY SUBROUTINE
; REG = R3,R4

mDELAY:     MOV   R3,#3FH
mDELAY1:    MOV   R4,#0
                  DJNZ  R4,$
        DJNZ  R3,mDELAY1
                  RET


;***** Sub Percent (0 - 100%) **********
;IN   LL_BUFH,V03,V04,V05,V06
;OUT  LCD

Percent:    CLR   C
    ;mov   ll_bufh,#7FH
    MOV   A,LL_BUFH    ;INPUT ******************************
    RLC   A
    MOV   R4,A
    LCALL LOOK_LL

    MOV   A,LL_BUF     ;***Take NEW VALUE to display***

        LCALL HTOA              ;
        MOV   A,V4              ;-X-.-%
        LCALL X1LCDWI           ;
        MOV   A,R2              ;
        LCALL X1LCDWD           ;
        MOV   A,V5              ;--X.-%
        LCALL X1LCDWI           ;
        MOV   A,R3              ;
        LCALL X1LCDWD           ;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    MOV   A,LL_BUFP    ;***Take NEW VALUE to display***

        LCALL  HTOA              ;
        mov r2,#0
        MOV   A,V6               ;---X%
        LCALL  X1LCDWI           ;
        MOV   A,R3               ;
        LCALL  X1LCDWD           ;


    MOV   A,LL_BUFH
    CJNE  A,#0FFH,LL_0
    MOV   R3,#31H     ;=1 (100.0)
    MOV   R2,#0
        MOV   A,V3               ;X--.-%
        LCALL  X1LCDWI           ;
        MOV   A,R3               ;
        LCALL  X1LCDWD           ;

        sjmp  Pout


LL_0:  MOV   R3,#30H     ;=0 (0XX.X)
    MOV   R2,#0
        MOV   A,V3               ;X--.-%
        LCALL  X1LCDWI           ;
        MOV   A,R3               ;
        LCALL  X1LCDWD           ;


Pout:       RET

;************ LOOK_LL SUB ***********
;INPUT=LL_BUFH
;OUTPUT=LL_BUFP,LL_BUF

LOOK_LL:
    MOV   R3,#02H     ;LOOP
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
LLL_0: JB    LL_BUFH.7,LLL_2
       MOV   A,R4
       MOV   DPTR,#TABLE0
LLL_3: MOVC  A,@A+DPTR
       CJNE  R3,#1,LLL_1
       MOV   LL_BUFP,A
       RET
LLL_2: MOV   A,R4
       MOV   DPTR,#TABLE1
       SJMP  LLL_3

LLL_1: MOV   LL_BUF,A
       INC   R4
       DJNZ  R3,LLL_0


;************* TABLE0 FOR HEX TO DEC ************

TABLE0: DB    00H,00H     ;0
        DB    00H,04H     ;1
        DB    00H,08H     ;2
        DB    01H,02H     ;3
        DB    01H,06H     ;4
        DB    02H,00H     ;5
        DB    02H,03H     ;6
        DB    02H,07H     ;7
        DB    03H,01H     ;8
        DB    03H,05H     ;9
        DB    03H,09H     ;10 **********
        DB    04H,03H     ;1
        DB    04H,07H     ;2
        DB    05H,01H     ;3
        DB    05H,05H     ;4
        DB    05H,09H     ;5
        DB    06H,03H     ;6
        DB    06H,07H     ;7
        DB    07H,01H     ;8
        DB    07H,05H     ;9
        DB    07H,08H     ;20 **********
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        DB    08H,02H     ;1
        DB    08H,06H     ;2
        DB    09H,00H     ;3
        DB    09H,04H     ;4
        DB    09H,08H     ;5
        DB    10H,02H     ;6
        DB    10H,06H     ;7
        DB    11H,00H     ;8
        DB    11H,04H     ;9
        DB    11H,08H     ;30 **********
        DB    12H,02H     ;1
        DB    12H,05H     ;2
        DB    12H,09H     ;3
        DB    13H,03H     ;4
        DB    13H,07H     ;5
        DB    14H,01H     ;6
        DB    14H,05H     ;7
        DB    14H,09H     ;8
        DB    15H,03H     ;9
        DB    15H,07H     ;40 **********
        DB    16H,01H     ;1
        DB    16H,05H     ;2
        DB    16H,09H     ;3
        DB    17H,03H     ;4
        DB    17H,07H     ;5
        DB    18H,00H     ;6
        DB    18H,04H     ;7
        DB    18H,08H     ;8
        DB    19H,02H     ;9
        DB    19H,06H     ;50 **********
        DB    20H,00H     ;1
        DB    20H,04H     ;2
        DB    20H,08H     ;3
        DB    21H,02H     ;4
        DB    21H,06H     ;5
        DB    22H,00H     ;6
        DB    22H,04H     ;7
        DB    22H,07H     ;8
        DB    23H,01H     ;9
        DB    23H,05H     ;60 **********
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
DB    23H,09H     ;1
DB    24H,03H     ;2
DB    24H,07H     ;3
DB    25H,01H     ;4
DB    25H,05H     ;5
DB    25H,09H     ;6
DB    26H,03H     ;7
DB    26H,07H     ;8
DB    27H,01H     ;9
DB    27H,05H     ;70 **********
DB    27H,09H     ;1
DB    28H,02H     ;2
DB    28H,06H     ;3
DB    29H,00H     ;4
DB    29H,04H     ;5
DB    29H,08H     ;6
DB    30H,02H     ;7
DB    30H,06H     ;8
DB    31H,00H     ;9
DB    31H,04H     ;80 **********
DB    31H,08H     ;1
DB    32H,02H     ;2
DB    32H,06H     ;3
DB    32H,09H     ;4
DB    33H,03H     ;5
DB    33H,07H     ;6
DB    34H,01H     ;7
DB    34H,05H     ;8
DB    34H,09H     ;9
DB    35H,03H     ;90 **********
DB    35H,07H     ;1
DB    36H,01H     ;2
DB    36H,05H     ;3
DB    36H,09H     ;4
DB    37H,03H     ;5
DB    37H,07H     ;6
DB    38H,00H     ;7
DB    38H,04H     ;8
DB    38H,08H     ;9
DB    39H,02H     ;100 *********
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        DB    39H,06H     ;1
        DB    40H,00H     ;2
        DB    40H,04H     ;3
        DB    40H,08H     ;4
        DB    41H,02H     ;5
        DB    41H,06H     ;6
        DB    42H,00H     ;7
        DB    42H,04H     ;8
        DB    42H,07H     ;9
        DB    43H,01H     ;110 *********
        DB    43H,05H     ;1
        DB    43H,09H     ;2
        DB    44H,03H     ;3
        DB    44H,07H     ;4
        DB    45H,01H     ;5
        DB    45H,05H     ;6
        DB    45H,09H     ;7
        DB    46H,03H     ;8
        DB    46H,07H     ;9
        DB    47H,01H     ;120 *********
        DB    47H,05H     ;1
        DB    47H,08H     ;2
        DB    48H,02H     ;3
        DB    48H,06H     ;4
        DB    49H,00H     ;5
        DB    49H,04H     ;6
        DB    49H,08H     ;7  127D=7FH=50%
TABLE1: DB    50H,02H     ;8
        DB    50H,06H     ;9
        DB    51H,00H     ;130 *********
        DB    51H,04H     ;1
        DB    51H,08H     ;2
        DB    52H,02H     ;3
        DB    52H,06H     ;4
        DB    53H,00H     ;5
        DB    53H,03H     ;6
        DB    53H,07H     ;7
        DB    54H,01H     ;8
        DB    54H,05H     ;9
        DB    54H,09H     ;140 *********
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
DB    55H,03H     ;1
DB    55H,07H     ;2
DB    55H,09H     ;3
DB    56H,05H     ;4
DB    56H,09H     ;5
DB    57H,03H     ;6
DB    57H,06H     ;7
DB    58H,00H     ;8
DB    58H,04H     ;9
DB    58H,08H     ;150 *********
DB    59H,02H     ;1
DB    59H,06H     ;2
DB    60H,00H     ;3
DB    60H,04H     ;4
DB    60H,08H     ;5
DB    61H,02H     ;6
DB    61H,06H     ;7
DB    62H,00H     ;8
DB    62H,04H     ;9
DB    62H,08H     ;160 *********
DB    63H,01H     ;1
DB    63H,05H     ;2
DB    63H,09H     ;3
DB    64H,03H     ;4
DB    64H,07H     ;5
DB    65H,01H     ;6
DB    65H,05H     ;7
DB    65H,09H     ;8
DB    66H,03H     ;9
DB    66H,07H     ;170 *********
DB    67H,01H     ;1
DB    67H,05H     ;2
DB    67H,08H     ;3
DB    68H,02H     ;4
DB    68H,06H     ;5
DB    69H,00H     ;6
DB    69H,04H     ;7
DB    69H,08H     ;8
DB    70H,02H     ;9
DB    70H,06H     ;180 *********
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        DB    71H,00H     ;1
        DB    71H,04H     ;2
        DB    71H,08H     ;3
        DB    72H,02H     ;4
        DB    72H,06H     ;5
        DB    72H,09H     ;6
        DB    73H,03H     ;7
        DB    73H,07H     ;8
        DB    74H,01H     ;9
        DB    74H,05H     ;190 *********
        DB    74H,09H     ;1
        DB    75H,03H     ;2
        DB    75H,07H     ;3
        DB    76H,01H     ;4
        DB    76H,05H     ;5
        DB    76H,09H     ;6
        DB    77H,03H     ;7
        DB    77H,07H     ;8
        DB    78H,01H     ;9
        DB    78H,04H     ;200 *********
        DB    78H,08H     ;1
        DB    79H,02H     ;2
        DB    79H,06H     ;3
        DB    80H,00H     ;4
        DB    80H,04H     ;5
        DB    80H,08H     ;6
        DB    81H,02H     ;7
        DB    81H,06H     ;8
        DB    82H,00H     ;9
        DB    82H,04H     ;210 *********
        DB    82H,08H     ;1
        DB    83H,01H     ;2
        DB    83H,05H     ;3
        DB    83H,09H     ;4
        DB    84H,03H     ;5
        DB    84H,07H     ;6
        DB    85H,01H     ;7
        DB    85H,05H     ;8
        DB    85H,09H     ;9
        DB    86H,02H     ;220 *********
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        DB    86H,07H     ;1
        DB    87H,01H     ;2
        DB    87H,05H     ;3
        DB    87H,09H     ;4
        DB    88H,02H     ;5
        DB    88H,06H     ;6
        DB    89H,00H     ;7
        DB    89H,04H     ;8
        DB    89H,08H     ;9
        DB    90H,02H     ;230 *********
        DB    90H,06H     ;1
        DB    91H,00H     ;2
        DB    91H,04H     ;3
        DB    91H,08H     ;4
        DB    92H,02H     ;5
        DB    92H,06H     ;6
        DB    92H,09H     ;7
        DB    93H,03H     ;8
        DB    93H,07H     ;9
        DB    94H,01H     ;240 *********
        DB    94H,05H     ;1
        DB    94H,09H     ;2
        DB    95H,03H     ;3
        DB    95H,07H     ;4
        DB    96H,01H     ;5
        DB    96H,05H     ;6
        DB    96H,09H     ;7
        DB    97H,03H     ;8
        DB    97H,07H     ;9
        DB    98H,00H     ;250 *********
        DB    98H,04H     ;1
        DB    98H,08H     ;2
        DB    99H,02H     ;3
        DB    99H,06H     ;4
        DB    00H,00H     ;5  255D=0FFH=100%

    ;-----------------------------------------
    ;---------------------- END ----------------------
    ;-----------------------------------------
    END
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
; FILENAME    PID.ASM
; DESCRIPTION  PID CONTROLLER
; HARDWARE    JAZZ-31
; ASSEMBLER   SXA51

; ********** VARIABLE **********
COM_PORT EQU   0FA00H ;READ-WRITE RES
RD_BUSY  EQU   0FA01H ;READ(BUSY FLAG)
DAT_PORT EQU   0FA02H ;WRITE CHARACTER
RD_DATA  EQU   0FA03H ;READ DATA FROM DD RAM

PORTA   EQU   0F800H ;INTPUT FROM PROCESS(PV)
PORTB   EQU   0F801H ;OUTPUT TO PROCESS(MV)
PORTC   EQU   0F802H ;INPUT FROM KEYBOARD
PORTP   EQU   0F803H ;8255 CONTROL PORT
RUN_STEP EQU   23H
BUF_LCD EQU   24H

HL_BUFH EQU   28H
LL_BUFH EQU   29H
SP_BUFH EQU   2AH
KP_BUFH EQU   2BH
TI_BUFH EQU   2CH
TD_BUFH EQU   2DH
PV_BUFH EQU   2EH
MV_BUFH EQU   2FH

HL_BUF  EQU   30H
HL_BUFP EQU   31H

LL_BUF  EQU   32H
LL_BUFP EQU   33H

SP_BUF  EQU   34H
SP_BUFP EQU   35H

TI_BUF0 EQU   36H
TI_BUF1 EQU   37H

TD_BUF0 EQU   38H
TD_BUF1 EQU   39H

PV_BUF  EQU   3AH
PV_BUFP EQU   3BH
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
MV_BUF  EQU  3CH
MV_BUFP EQU  3DH

DTI_BUFH EQU  3FH
DTD_BUFH EQU  40H
EN_BUFH  EQU  41H
EN_1_BUFH EQU  42H
TD_DTD  EQU  43H

;********** Dim Variable ***********

V0   EQU  50H
V1   EQU  51H
V2   EQU  52H
V3   EQU  53H
V4   EQU  54H
V5   EQU  55H
V6   EQU  56H
V7   EQU  57H

V_BlockST   EQU  60H
V_Block00   EQU  61H
V_Block01   EQU  62H
V_Block02   EQU  63H
V_Block03   EQU  64H
V_Block04   EQU  65H
V_Block05   EQU  66H
V_Block06   EQU  67H
V_Block07   EQU  68H
V_BlockEND  EQU  69H

;*****************************;
;   SERIAL USE INT
;      V-31
;*****************************;
    ORG  0000H
    LJMP  ST

    ORG  0023H      ;REM31 INT Vector RI+TI : MCS-51(0023H)
    JBC  RI,RECV    ;TI+RI Vector Action
    LJMP  OUT       ;if TI reti OUT

RECV:
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        PUSH   00H
        PUSH   01H
        PUSH   02H
        PUSH   03H
        PUSH   04H
        PUSH   05H
        PUSH   06H
        PUSH   07H
        PUSH   ACC
        PUSH   PSW
        PUSH   20H
        PUSH   21H
        PUSH   22H

        LCALL  RxBlock      ;Save MV to V_Block
        MOV    A,V_BlockST
        CJNE   A,#21H ,OUT   ;!
        MOV    A,V_Block00
        CJNE   A,#30H ,OUT   ;0
        MOV    A,V_Block01
        CJNE   A,#32H ,OUT   ;2

        LCALL  MVO          ;Get MV from Block,Convert to HEX,OUT PORT1

        LCALL  SEND
        JNB    TI,$
        CLR    TI
        SETB   ES

OUT:
        POP    22H
        POP    21H
        POP    20H
        POP    PSW
        POP    ACC
        POP    07H
        POP    06H
        POP    05H
        POP    04H
        POP    03H
        POP    02H
        POP    01H
        POP    00H
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    RETI

SEND:
    SETB  P3.4
    LCALL  PVIB        ;IN PV,Save to V_Block *** INPUT HERE ***

    LCALL  CHKSUM
    MOV   V_Block07,A
    LCALL  TxBlock  ;Tx V_Block
    CLR   P3.4
    RET

    ORG   0100H
;****************** Set Const *****************
ST:  LCALL  INIv
    CLR   P3.4

OVER:
    MOV   SCON,#50H    ;Serial Mode 1 9600
    MOV   TMOD,#20H
    MOV   TH1,#0FDH
    SETB  TR1
    SETB  EA
    SETB  ES

    LJMP  RES

INIv:
    MOV   V_BlockST,#00100001B  ;!
    MOV   V_BlockEND,#00100011B  ;#

    MOV   V_Block00,#30H      ;0   ST.No.
    MOV   V_Block01,#32H      ;2   ST.No.
    MOV   V_Block02,#57H      ;W   General Command & Status
    MOV   V_Block03,#2BH      ;+   DATA Send PV
    MOV   V_Block04,#2BH      ;+   & Recieve SP
    MOV   V_Block05,#2BH      ;+   DATA Send MV
    MOV   V_Block06,#2BH      ;+
    MOV   V_Block07,#24H      ;$   ChkSUM

    RET

;************** TxBlock ************
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
TxBlock:MOV   R0,#60H
TxBLoop:MOV   A,@R0
    LCALL  Tx1Byte
    INC   R0
    CJNE   R0,#6AH,TxBLoop
    RET

; ********** Tx1Byte SUB **********
; SEND 1 BYTE
; IN = A
; REG = NO

Tx1Byte:CLR   TI
    CLR   ES
    MOV   SBUF,A
    JNB   TI,$
    RET

;************** RxBlock ************
RxBlock:
    LCALL  Rx1Byte
    MOV   V_BlockST,A   ;SAVE 1ST from PC TO BLOCK
    CJNE   A,#21H,RxOUT   ;!

    LCALL  Rx1Byte      ;ST.No. Byte 1
    MOV   V_Block00,A   ;SAVE from PC TO BLOCK
    CJNE   A,#30H,RxOUT   ;0

    LCALL  Rx1Byte      ;ST.No. Byte 2
    MOV   V_Block01,A   ;SAVE from PC TO BLOCK
    CJNE   A,#32H,RxOUT   ;2

    LCALL  Rx1Byte      ;GSC Byte

    LCALL  Rx1Byte
    MOV   V_Block03,A   ;SAVE MV from PC TO BLOCK

    LCALL  Rx1Byte
    MOV   V_Block04,A   ;SAVE MV from PC TO BLOCK

    LCALL  Rx1Byte      ;+
    LCALL  Rx1Byte      ;+
    LCALL  Rx1Byte      ;ChkSum
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
RxOUT: RET

; ********** Rx1Byte SUB **********

Rx1Byte:MOV   A,SBUF
    JNB   RI,$          ;WAIT FOR RECEIVE OK
    CLR   RI
    RET

;***********PV IN & to V_Block SUB************

PVIB:

    MOV   A,PV_BUFH     ;Get input
    MOV   V1,A          ;SAVE INPUT TO V1
    MOV   A,V1
    LCALL  HTOA
    MOV   V_Block03,R2   ;SAVE PV TO V_BLOCK
    MOV   V_Block04,R3   ;SAVE PV TO V_BLOCK

    MOV   A,MV_BUFH     ;Get input
    MOV   V1,A          ;SAVE INPUT TO V1
    MOV   0,V1
    LCALL  HTOA
    MOV   V_Block05,R2   ;SAVE MV TO V_BLOCK
    MOV   V_Block06,R3   ;SAVE MV TO V_BLOCK

    RET

;***********MV OUT SUB************

MVO:

    MOV   R2,V_Block03   ;SP from PC TO Controller
    MOV   R3,V_Block04
    LCALL  ATOH          ;Convert to HEX
    MOV   V2,A          ;Save to V2

    MOV   A,V2          ;SAVE SP FROM PC TO SP_BUFH
    MOV   SP_BUFH,A
    RET

; ********** ATOH SUB **********
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
; ASCII TO HEX CONVERT
; IN  = R2,R3  30H,41H
; OUT = A     0AH
; REG = A,R2

ATOH:  MOV    A,R2
       LCALL  ATOHS
       SWAP   A
       MOV    R2,A
       MOV    A,R3
       LCALL  ATOHS
       ORL    A,R2
       RET

ATOHS: CJNE   A,#'A',$+3
       JC     ATOHS1
       ADD    A,#9

ATOHS1: ANL   A,#0FH
       RET

; ********** CHKSUM SUB **********
;Sum V_Block00-06 without carry
;IN = V_Block00-06
;OUT = A

CHKSUM:
       MOV    A,#00H
       ADD    A,V_Block00
       ADD    A,V_Block01
       ADD    A,V_Block02
       ADD    A,V_Block03
       ADD    A,V_Block04
       ADD    A,V_Block05
       ADD    A,V_Block06

       RET

; ********** RESET ************
; SYSTEM RESET

RES:   MOV    R2,#40H        ;POWER UP DELAY
RES1:  MOV    R3,#0
       DJNZ   R3,$
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    DJNZ  R2,RES1

    MOV   R2,#3FH
    MOV   R0,#20H
XX: MOV   @R0,#0
    INC   R0
    DJNZ  R2,XX


START: MOV   DPTR,#PORTP
    MOV   A,#99H      ;PA=INPUT,PB=OUTPUT,PC=INPUT
    MOVX  @DPTR,A
    CLR   A

;******** LCD INITIALIZE ********
    MOV   A,#00111000B  ;FUNCTION SET
    LCALL WR_COMM       ;DL=1,N=1,F=0
    MOV   A,#00001100B  ;DISPLAY ON/OFF
    LCALL WR_COMM       ;D=1,C=0,B=0
    MOV   A,#01H        ;CLEAR
    LCALL WR_COMM

    MOV   DPTR,#TITLE   ;INSTRUMENTATION ENGINEERING
    LCALL DSPLY
    MOV   DPTR,#LOGO    ;PID CONTROLLER..
    LCALL DSPLY
    MOV   DPTR,#NAME0
    LCALL DSPLY
    MOV   DPTR,#NAME1
    LCALL DSPLY
    SETB  20H.0         ;CHOOSE MENU_1
    SETB  21H.0         ;CHOOSE MAN
    SETB  20H.6         ;CHOOSE DIR
    SJMP  MAIN

DSPLY: LCALL  DISPLAY
    LCALL DELAY
    RET

;************ MAIN ************
MAIN: JB    20H.0,A_1    ;MENU_1
    JB    20H.1,A_2    ;MENU_2
    JB    20H.2,A_3    ;MENU_3
    JB    20H.3,A_4    ;MENU_4
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    JB    20H.4,A_5    ;MENU_5
    JB    20H.5,A_6    ;MENU_6

A_1:  LJMP   MENU_1
A_2:  LJMP   MENU_2
A_3:  LJMP   MENU_3
A_4:  LJMP   MENU_4
A_5:  LJMP   MENU_5
A_6:  LJMP   MENU_6


MAIN1: LCALL  INPUT
    CJNE   A,#1,MAIN
    LJMP   A_2
MAIN2: LCALL  INPUT
    CJNE   A,#1,MAIN
    LJMP   A_3
MAIN3: LCALL  INPUT
    CJNE   A,#1,MAIN
    LJMP   A_4
MAIN4: LCALL  INPUT
    CJNE   A,#1,MAIN
    LJMP   A_5
MAIN5: LCALL  INPUT
    CJNE   A,#1,MAIN
    LJMP   A_6
MAIN6: LCALL  INPUT
    CJNE   A,#1,MAIN
    LJMP   A_1

;************ MENU_1 SUB ********
MENU_1: CLR    20H.1
    CLR    20H.2
    CLR    20H.3
    CLR    20H.4
    CLR    20H.5
    SETB   20H.0
;**** DEMO SUB ****
DEMO:  JB    21H.0,A_7    ;MAN
    JB    21H.1,A_8    ;P
    JB    21H.2,A_9    ;PI
    JB    21H.3,A_10   ;PID
    JB    21H.4,A_11   ;PD
    JB    21H.5,A_12   ;ONOF
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
A_7:  LJMP   DEMO_1
A_8:  LJMP   DEMO_2
A_9:  LJMP   DEMO_3
A_10: LJMP   DEMO_4
A_11: LJMP   DEMO_5
A_12: LJMP   DEMO_6


DEMO1: LCALL  INPUT
    CJNE   A,#2,X_0
    LJMP   A_8
DEMO2: LCALL  INPUT
    CJNE   A,#2,X_0
    LJMP   A_9
DEMO3: LCALL  INPUT
    CJNE   A,#2,X_0
    LJMP   A_10
DEMO4: LCALL  INPUT
    CJNE   A,#2,X_0
    LJMP   A_11
DEMO5: LCALL  INPUT
    CJNE   A,#2,X_0
    LJMP   A_12
DEMO6: LCALL  INPUT
    CJNE   A,#2,X_0
    LJMP   A_7
X_0:  LJMP   DIS_SP      ;MAIN1

;**** DEMO_X_X SUB ****
DEMO_1: CLR   21H.1
    CLR   21H.2
    CLR   21H.3
    CLR   21H.4
    CLR   21H.5
    SETB  21H.0       ;MAN
    JB    20H.6,A_13   ;DIR
    JNB   20H.6,A_14   ;REV
A_13: LJMP   M_DIR
A_14: LJMP   M_REV

M_D_1: LCALL  INPUT
    CJNE   A,#4,X_1
    LJMP   A_14
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
M_R_1: LCALL  INPUT
    CJNE   A,#4,X_1
    LJMP   A_13
X_1:  LJMP   DEMO1

M_DIR: SETB   20H.6       ;DIR
    MOV    DPTR,#DEMO_M_D
    LCALL  DISPLAY
    LJMP   M_D_1
M_REV: CLR    20H.6       ;REV
    MOV    DPTR,#DEMO_M_R
    LCALL  DISPLAY
    LJMP   M_R_1

DEMO_2: CLR   21H.0
    CLR    21H.2
    CLR    21H.3
    CLR    21H.4
    CLR    21H.5
    SETB   21H.1       ;P
    JB     20H.6,A_15   ;DIR
    JNB    20H.6,A_16   ;REV
A_15:  LJMP   P_DIR
A_16:  LJMP   P_REV

P_D_1: LCALL  INPUT
    CJNE   A,#4,X_2
    LJMP   A_16
P_R_1: LCALL  INPUT
    CJNE   A,#4,X_2
    LJMP   A_15
X_2:  LJMP   DEMO2

P_DIR: SETB   20H.6       ;DIR
    MOV    DPTR,#DEMO_P_D
    LCALL  DISPLAY
    LJMP   P_D_1
P_REV: CLR    20H.6       ;REV
    MOV    DPTR,#DEMO_P_R
    LCALL  DISPLAY
    LJMP   P_R_1

DEMO_3: CLR   21H.0
    CLR    21H.1
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    CLR   21H.3
    CLR   21H.4
    CLR   21H.5
    SETB  21H.2      ;PI
    JB    20H.6,A_17   ;DIR
    JNB   20H.6,A_18   ;REV
A_17: LJMP  PI_DIR
A_18: LJMP  PI_REV

PI_D_1: LCALL  INPUT
    CJNE  A,#4,X_3
    LJMP  A_18
PI_R_1: LCALL  INPUT
    CJNE  A,#4,X_3
    LJMP  A_17
X_3:  LJMP  DEMO3

PI_DIR: SETB  20H.6      ;DIR
    MOV   DPTR,#DEMO_PI_D
    LCALL  DISPLAY
    LJMP  PI_D_1
PI_REV: CLR   20H.6      ;REV
    MOV   DPTR,#DEMO_PI_R
    LCALL  DISPLAY
    LJMP  PI_R_1

DEMO_4: CLR   21H.0
    CLR   21H.1
    CLR   21H.2
    CLR   21H.4
    CLR   21H.5
    SETB  21H.3      ;PID
    JB    20H.6,A_19   ;DIR
    JNB   20H.6,A_20   ;REV
A_19: LJMP  PID_DIR
A_20: LJMP  PID_REV

PID_D_1:LCALL  INPUT
    CJNE  A,#4,X_4
    LJMP  A_20
PID_R_1:LCALL  INPUT
    CJNE  A,#4,X_4
    LJMP  A_19
X_4:  LJMP  DEMO4
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
PID_DIR:SETB  20H.6      ;DIR
    MOV   DPTR,#DEMO_PID_D
    LCALL  DISPLAY
    LJMP   PID_D_1
PID_REV:CLR   20H.6      ;REV
    MOV   DPTR,#DEMO_PID_R
    LCALL  DISPLAY
    LJMP   PID_R_1

DEMO_5: CLR   21H.0
    CLR   21H.1
    CLR   21H.2
    CLR   21H.3
    CLR   21H.5
    SETB  21H.4      ;PD
    JB    20H.6,A_21   ;DIR
    JNB   20H.6,A_22   ;REV
A_21:  LJMP   PD_DIR
A_22:  LJMP   PD_REV

PD_D_1: LCALL  INPUT
    CJNE  A,#4,X_5
    LJMP   A_22
PD_R_1: LCALL  INPUT
    CJNE  A,#4,X_5
    LJMP   A_21
X_5:   LJMP   DEMO5

PD_DIR: SETB   20H.6      ;DIR
    MOV   DPTR,#DEMO_PD_D
    LCALL  DISPLAY
    LJMP   PD_D_1
PD_REV: CLR   20H.6      ;REV
    MOV   DPTR,#DEMO_PD_R
    LCALL  DISPLAY
    LJMP   PD_R_1

DEMO_6: CLR   21H.0
    CLR   21H.1
    CLR   21H.2
    CLR   21H.3
    CLR   21H.4
    SETB  21H.5      ;ONOF
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    JB    20H.6,A_23    ;DIR
    JNB   20H.6,A_24    ;REV
A_23: LJMP  ONOF_DIR
A_24: LJMP  ONOF_REV

ONOF_D_1:
    LCALL  INPUT
    CJNE   A,#4,X_6
    LJMP   A_24
ONOF_R_1:
    LCALL  INPUT
    CJNE   A,#4,X_6
    LJMP   A_23
X_6:  LJMP  DEMO6

ONOF_DIR:
    SETB   20H.6       ;DIR
    MOV    DPTR,#DEMO_ONOF_D
    LCALL  DISPLAY
    LJMP   ONOF_D_1
ONOF_REV:
    CLR    20H.6       ;REV
    MOV    DPTR,#DEMO_ONOF_R
    LCALL  DISPLAY
    LJMP   ONOF_R_1

;********* DIS_SP SUB **********
DIS_SP: CLR   C
    MOV    A,SP_BUFH
    RLC    A
    MOV    R4,A
    LCALL  LOOK_SP

    MOV    DPTR,#PORTA
    MOVX   A,@DPTR
    MOV    PV_BUFH,A
    LCALL  DIS_PV

    MOV    A,SP_BUF     ;***Take NEW VALUE to display***
    LCALL  HTOA
    MOV    BUF_LCD,R2
    MOV    BUF_LCD+1,R3
    MOV    RUN_STEP,#84H
    MOV    R1,#BUF_LCD
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        MOV   BUF_LCD+3,R3
        MOV   RUN_STEP,#0CAH
        MOV   R1,#BUF_LCD+3
        LCALL  RUN_2R11

        LJMP  JK25

L_0:  CLR   C
        MOV   A,PV_BUFH
        SUBB  A,LL_BUFH
        JZ    L_1       ;PV=LL DISPLAY
        JC    L_I       ;PV<LL DISPLAY
        LJMP  H_L_1

L_1:  MOV   R3,#4CH     ;=L
        MOV   R2,#0
        MOV   BUF_LCD+3,R3
        MOV   RUN_STEP,#0C9H
        MOV   R1,#BUF_LCD+3
        LCALL  RUN_2R11

        MOV   R3,#4CH     ;=L
        MOV   R2,#0
        MOV   BUF_LCD+3,R3
        MOV   RUN_STEP,#0CAH
        MOV   R1,#BUF_LCD+3
        LCALL  RUN_2R11
        LJMP  JK25

H_L_1: MOV   R3,#0A0H     ;=EMPTLY
        MOV   R2,#0
        MOV   BUF_LCD+3,R3
        MOV   RUN_STEP,#0C9H
        MOV   R1,#BUF_LCD+3
        LCALL  RUN_2R11

        MOV   R3,#0A0H     ;=0
        MOV   R2,#0
        MOV   BUF_LCD+3,R3
        MOV   RUN_STEP,#0CAH
        MOV   R1,#BUF_LCD+3
        LCALL  RUN_2R11

JK25:  LCALL  INPUT1
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    CJNE  A,#18H,SP_1
    SETB  F0
    LCALL SP_INC

SP_1: LCALL INPUT1
    MOV   R6,A
    CJNE  A,#10H,JK26
    LCALL SP_INC

JK26: LCALL INPUT1
    CJNE  A,#28H,SP_2
    SETB  F0
    LCALL SP_DEC

SP_2: MOV   A,R6
    CJNE  A,#20H,SP_3
    LCALL SP_DEC
SP_3: LCALL INPUT1
    MOV   R2,A
    MOV   R3,A
    CJNE  A,#1,SP_4
    LJMP  MAIN1
SP_4: CJNE  R2,#2,SP_5
    SJMP  SP_6
SP_5: CJNE  R3,#4,SP_13
SP_6: JB    21H.0,SP_7    ;MAN
    JB    21H.1,SP_8    ;P
    JB    21H.2,SP_9    ;PI
    JB    21H.3,SP_10   ;PID
    JB    21H.4,SP_11   ;PD
    JB    21H.5,SP_12   ;ONOF
SP_7: LJMP  DEMO_1
SP_8: LJMP  DEMO_2
SP_9: LJMP  DEMO_3
SP_10: LJMP  DEMO_4
SP_11: LJMP  DEMO_5
SP_12: LJMP  DEMO_6

SP_13: LJMP  MAIN_CON

SP_INC:              ;SP INCREMENT
    MOV   A,SP_BUFH
    RLC   A
    MOV   R4,A
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    LCALL  LOOK_SP
    JBC    F0,JK27
    LCALL  DELAY_SL
    SJMP   JK28
JK27: LCALL  DELAY_Q
JK28: MOV    A,SP_BUFH
    CJNE   A,#0FFH,SP_14
    RET
SP_14: INC   SP_BUFH
    RET

SP_DEC:              ;SP DECREMENT
    MOV    A,SP_BUFH
    RLC    A
    MOV    R4,A
    LCALL  LOOK_SP
    JBC    F0,JK29
    LCALL  DELAY_SL
    SJMP   JK30
JK29: LCALL  DELAY_Q
JK30: MOV    A,SP_BUFH
    CJNE   A,#0H,SP_15
    RET
SP_15: DEC   SP_BUFH
    RET

;********** MENU_2 SUB *******
MENU_2: CLR    20H.0        ;KP
    CLR    20H.2
    CLR    20H.3
    CLR    20H.4
    CLR    20H.5
    SETB   20H.1
    MOV    DPTR,#MENU_2D
    LCALL  DISPLAY
    LCALL  DIS_KP
    LJMP   MAIN2

;********** DIS_KP SUB *********
DIS_KP:
    MOV    A,KP_BUFH     ;***Take NEW VALUE to display***
    LCALL  HTOA
    MOV    BUF_LCD,R2
    MOV    BUF_LCD+1,R3
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    MOV   RUN_STEP,#0CDH
    MOV   R1,#BUF_LCD
    LCALL  RUN_2R1      ;*********************************

    LCALL  INPUT1
    MOV   R6,A
    CJNE   A,#10H,KP_1
    MOV   A,KP_BUFH
    CJNE   A,#9,KP_0     ;IF KP_BUFH = 9 NOT INCREMENT
    LJMP   DIS_KP

KP_0:  INC    KP_BUFH
    LCALL  DELAY2
    LJMP   DIS_KP
KP_1:  MOV    A,R6
    CJNE   A,#20H,KP_2
    MOV   A,KP_BUFH
    CJNE   A,#0,KP_3     ;IF KP_BUFH = 0 NOT DECREMENT
    LJMP   DIS_KP
KP_3:  DEC    KP_BUFH
    LCALL  DELAY2
    LJMP   DIS_KP
KP_2:  MOV    A,R6
    RET

;*********** MENU_3 SUB *******
MENU_3: CLR    20H.0
    CLR   20H.1
    CLR   20H.3
    CLR   20H.4
    CLR   20H.5
    SETB  20H.2
    MOV   DPTR,#MENU_3D
    LCALL  DISPLAY

;*********** TI_BUFH SUB *********
DIS_TI:
    MOV   A,TI_BUF0     ;***Take NEW VALUE to display***
    LCALL  HTOA
    MOV   BUF_LCD,R2
    MOV   BUF_LCD+1,R3
    MOV   RUN_STEP,#0CDH
    MOV   R1,#BUF_LCD
    LCALL  RUN_2R1      ;*********************************
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    MOV   A,TI_BUF1     ;***Take NEW VALUE to display***
    LCALL  HTOA
    MOV   BUF_LCD+2,R2
    MOV   BUF_LCD+3,R3
    MOV   RUN_STEP,#0CFH
    MOV   R1,#BUF_LCD+2
    LCALL  RUN_2R1      ;*******************************

    LCALL  INPUT1
    CJNE   A,#18H,TI_1
    SETB   F0
    LJMP   TI_INC

TI_1: LCALL  INPUT1
    MOV   R6,A
    CJNE   A,#10H,JK13
    LJMP   TI_INC

JK13: LCALL  INPUT1
    CJNE   A,#28H,TI_2
    SETB   F0
    LJMP   TI_DEC

TI_2: MOV   A,R6
    CJNE   A,#20H,TI_3
    LJMP   TI_DEC
TI_3: MOV   A,R6
    LCALL  INPUT1
    CJNE   A,#1,TI_4
    LJMP   MAIN3
TI_4: LJMP   DIS_TI

TI_INC:             ;TI INCREMENT
    MOV   A,TI_BUFH
    RLC   A
    MOV   R4,A
    LCALL  LOOK_TI
    JBC   F0,JK20
    LCALL  DELAY_SL
    SJMP   JK21
JK20: LCALL  DELAY_Q
JK21: MOV   A,TI_BUFH
    CJNE   A,#0FFH,TI_5
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    LJMP   DIS_TI
TI_5:  INC    TI_BUFH
    LJMP   DIS_TI

TI_DEC:              ;TI INCREMENT
    MOV    A,TI_BUFH
    RLC    A
    MOV    R4,A
    LCALL  LOOK_TI
    JBC    F0,JK22
    LCALL  DELAY_SL
    SJMP   JK23
JK22:  LCALL  DELAY_Q
JK23:  MOV    A,TI_BUFH
    CJNE   A,#0,TI_6
    LJMP   DIS_TI
TI_6:  DEC    TI_BUFH
    LJMP   DIS_TI

;********** MENU_4 SUB *******
MENU_4: CLR    20H.0
    CLR    20H.1
    CLR    20H.2
    CLR    20H.4
    CLR    20H.5
    SETB   20H.3
    MOV    DPTR,#MENU_4D
    LCALL  DISPLAY
;********** TD_BUFH SUB *********
DIS_TD:
    MOV    A,TD_BUF0     ;***Take NEW VALUE to display***
    LCALL  HTOA
    MOV    BUF_LCD,R2
    MOV    BUF_LCD+1,R3
    MOV    RUN_STEP,#0CDH
    MOV    R1,#BUF_LCD
    LCALL  RUN_2R1       ;*********************************

    MOV    A,TD_BUF1     ;***Take NEW VALUE to display***
    LCALL  HTOA
    MOV    BUF_LCD+2,R2
    MOV    BUF_LCD+3,R3
    MOV    RUN_STEP,#0CFH
    MOV    R1,#BUF_LCD+2
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    LCALL  RUN_2R1     ;*****************************

    LCALL  INPUT1
    CJNE   A,#18H,TD_1
    SETB   F0
    LJMP   TD_INC

TD_1: LCALL  INPUT1
    MOV    R6,A
    CJNE   A,#10H,JK18
    LJMP   TD_INC

JK18: LCALL  INPUT1
    CJNE   A,#28H,TD_2
    SETB   F0
    LJMP   TD_DEC

TD_2: MOV    A,R6
    CJNE   A,#20H,TD_3
    LJMP   TD_DEC
TD_3: MOV    A,R6
    LCALL  INPUT1
    CJNE   A,#1,TD_4
    LJMP   MAIN4
TD_4: LJMP   DIS_TD

TD_INC:              ;TD INCREMENT
    MOV    A,TD_BUFH
    RLC    A
    MOV    R4,A
    LCALL  LOOK_TD
    JBC    F0,JK14
    LCALL  DELAY_SL
    SJMP   JK15
JK14: LCALL  DELAY_Q
JK15: MOV    A,TD_BUFH
    CJNE   A,#0FFH,TD_5
    LJMP   DIS_TD
TD_5: INC    TD_BUFH
    LJMP   DIS_TD

TD_DEC:              ;TD DECREMENT
    MOV    A,TD_BUFH
    RLC    A
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        MOV   R4,A
        LCALL  LOOK_TD
        JBC   F0,JK16
        LCALL  DELAY_SL
        SJMP   JK17
JK16:  LCALL  DELAY_Q
JK17:  MOV    A,TD_BUFH
        CJNE   A,#0,TD_6
        LJMP   DIS_TD
TD_6:  DEC    TD_BUFH
        LJMP   DIS_TD

;********** MENU_5 SUB *******
MENU_5: CLR    20H.0
        CLR    20H.1
        CLR    20H.2
        CLR    20H.3
        CLR    20H.5
        SETB   20H.4
        MOV    DPTR,#MENU_5D
        LCALL  DISPLAY
;********** H_ALARM SUB *********
H_ALARM:CLR    C
        MOV    A,HL_BUFH
        RLC    A
        MOV    R4,A
        LCALL  LOOK_HL

        MOV    A,HL_BUF      ;***Take NEW VALUE to display***
        LCALL  HTOA
        MOV    BUF_LCD,R2
        MOV    BUF_LCD+1,R3
        MOV    RUN_STEP,#0CEH
        MOV    R1,#BUF_LCD
        LCALL  RUN_2R1       ;*******************************

        MOV    A,HL_BUFP     ;***Take NEW VALUE to display***
        LCALL  HTOA
        MOV    R2,#0
        MOV    BUF_LCD+2,R3
        MOV    RUN_STEP,#0D1H
        MOV    R1,#BUF_LCD+2
        LCALL  RUN_2R11      ;*******************************
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        MOV   A,HL_BUFH
        CJNE  A,#0FFH,HL_0
        MOV   R3,#31H      ;=1 (100.0)
        MOV   R2,#0
        MOV   BUF_LCD+3,R3
        MOV   RUN_STEP,#0CDH
        MOV   R1,#BUF_LCD+3
        LCALL RUN_2R11
        SJMP  JK0          ;HL_1

HL_0:   MOV   R3,#30H      ;=0 (0XX.X)
        MOV   R2,#0
        MOV   BUF_LCD+3,R3
        MOV   RUN_STEP,#0CDH
        MOV   R1,#BUF_LCD+3
        LCALL RUN_2R11

JK0:    LCALL INPUT1
        CJNE  A,#18H,HL_1
        SETB  F0
        LJMP  HL_INC

HL_1:   LCALL INPUT1
        MOV   R6,A
        CJNE  A,#10H,JK3
        LJMP  HL_INC

JK3:    LCALL INPUT1
        CJNE  A,#28H,HL_2
        SETB  F0
        LJMP  HL_DEC

HL_2:   MOV   A,R6
        CJNE  A,#20H,HL_3
        LJMP  HL_DEC

HL_3:   MOV   A,R6
        LCALL INPUT1
        CJNE  A,#1,HL_4
        LJMP  MAIN5
HL_4:   LJMP  H_ALARM

HL_INC: CLR   A            ;HL INCREMENT
        MOV   A,HL_BUFH
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    RLC   A
    MOV   R4,A
    LCALL  LOOK_HL
    JBC   F0,JK1
    LCALL  DELAY_SL
    SJMP   JK2
JK1:  LCALL  DELAY_Q
JK2:  MOV   A,HL_BUFH
    CJNE   A,#0FFH,HL_5
    LJMP   H_ALARM
HL_5:  INC   HL_BUFH
    LJMP   H_ALARM

HL_DEC: CLR   C          ;HL DECREMENT
    MOV   A,HL_BUFH
    RLC   A
    MOV   R4,A
    LCALL  LOOK_HL
    JBC   F0,JK4
    LCALL  DELAY_SL
    SJMP   JK5
JK4:  LCALL  DELAY_Q
JK5:  MOV   A,HL_BUFH
    CJNE   A,#0,HL_6
    LJMP   H_ALARM
HL_6:  DEC   HL_BUFH
    LJMP   H_ALARM

;********** MENU_6 SUB *******
MENU_6: CLR   20H.0
    CLR   20H.1
    CLR   20H.2
    CLR   20H.3
    CLR   20H.4
    SETB   20H.5
    MOV   DPTR,#MENU_6D
    LCALL  DISPLAY

;********** L_ALARM SUB *********
L_ALARM:
    CLR   C
    MOV   A,LL_BUFH
    RLC   A
    MOV   R4,A
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        LCALL  LOOK_LL

        MOV    A,LL_BUF      ;***Take NEW VALUE to display***
        LCALL  HTOA
        MOV    BUF_LCD,R2
        MOV    BUF_LCD+1,R3
        MOV    RUN_STEP,#0CEH
        MOV    R1,#BUF_LCD
        LCALL  RUN_2R1       ;*********************************

        MOV    A,LL_BUFP     ;***Take NEW VALUE to display***
        LCALL  HTOA
        MOV    R2,#0
        MOV    BUF_LCD+2,R3
        MOV    RUN_STEP,#0D1H
        MOV    R1,#BUF_LCD+2
        LCALL  RUN_2R11      ;*********************************

        MOV    A,LL_BUFH
        CJNE   A,#0FFH,LL_0
        MOV    R3,#31H       ;=1 (100.0)
        MOV    R2,#0
        MOV    BUF_LCD+3,R3
        MOV    RUN_STEP,#0CDH
        MOV    R1,#BUF_LCD+3
        LCALL  RUN_2R11
        SJMP   JK9

LL_0:   MOV    R3,#30H       ;=0 (0XX.X)
        MOV    R2,#0
        MOV    BUF_LCD+3,R3
        MOV    RUN_STEP,#0CDH
        MOV    R1,#BUF_LCD+3
        LCALL  RUN_2R11

JK9:    LCALL  INPUT1
        CJNE   A,#18H,LL_1
        SETB   F0
        LJMP   LL_INC

LL_1:   LCALL  INPUT1
        MOV    R6,A
        CJNE   A,#10H,JK6
        LJMP   LL_INC
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
JK6:  LCALL  INPUT1
      CJNE   A,#28H,LL_2
      SETB   F0
      LJMP   LL_DEC

LL_2: MOV    A,R6
      CJNE   A,#20H,LL_3
      LJMP   LL_DEC
LL_3: MOV    A,R6
      LCALL  INPUT1
      CJNE   A,#1,LL_4
      LJMP   MAIN6
LL_4: LJMP   L_ALARM

LL_INC: CLR  C          ;LL INCREMENT
      MOV    A,LL_BUFH
      RLC    A
      MOV    R4,A
      LCALL  LOOK_LL
      JBC    F0,JK10
      LCALL  DELAY_SL
      SJMP   JK11
JK10: LCALL  DELAY_Q
JK11: MOV    A,LL_BUFH
      CJNE   A,#0FFH,LL_5
      LJMP   L_ALARM
LL_5: INC    LL_BUFH
      LJMP   L_ALARM

LL_DEC: CLR  C          ;LL DECREMENT
      MOV    A,LL_BUFH
      RLC    A
      MOV    R4,A
      LCALL  LOOK_LL
      JBC    F0,JK7
      LCALL  DELAY_SL
      SJMP   JK8
JK7:  LCALL  DELAY_Q
JK8:  MOV    A,LL_BUFH
      CJNE   A,#0,LL_6
      LJMP   L_ALARM
LL_6: DEC    LL_BUFH
      LJMP   L_ALARM
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
;********** DISPLAY SUB ************
TITLE:      DB   "INSTRUMENTATION   "
            DB   "     ENGINEERING"
LOGO:       DB   "  PID CONTROLLER "
            DB   "  VERSION 1.0   "
NAME0:      DB   "     BY      "
            DB   "  T. JUKGRIT   "
NAME1:      DB   "  T. VERACHART  "
            DB   "  L. DUSSADEE   "
MENU_1D:    DB   "SP=000.0% MV=000.0% "
            DB   "PV=000.0%       "
MENU_2D:    DB   " PROPORTIONAL GAIN "
            DB   "      KP=00    "
MENU_3D:    DB   " INTREGRAL TIME   "
            DB   "      TI=0000 ms"
MENU_4D:    DB   " DEREVATIVE TIME  "
            DB   "      TD=0000 ms"
MENU_5D:    DB   " HIGH ALARM     "
            DB   "      HL=000.0% "
MENU_6D:    DB   " LOW ALARM      "
            DB   "      LL=000.0% "

;********* DEMO1 SUB *********;
DEMO_M_D:    DB   "SP=000.0% MV=000.0%"
            DB   "PV=000.0% MAN  DIR"
DEMO_P_D:    DB   "SP=000.0% MV=000.0%"
            DB   "PV=000.0% P    DIR"
DEMO_PI_D:   DB   "SP=000.0% MV=000.0%"
            DB   "PV=000.0% PI   DIR"
DEMO_PID_D:  DB   "SP=000.0% MV=000.0%"
            DB   "PV=000.0% PID  DIR"
DEMO_PD_D:   DB   "SP=000.0% MV=000.0%"
            DB   "PV=000.0% PD   DIR"
DEMO_ONOF_D: DB   "SP=000.0% MV=000.0%"
            DB   "PV=000.0% ONOFF DIR"
DEMO_M_R:    DB   "SP=000.0% MV=000.0%"
            DB   "PV=000.0% MAN  REV"
DEMO_P_R:    DB   "SP=000.0% MV=000.0%"
            DB   "PV=000.0% P    REV"
DEMO_PI_R:   DB   "SP=000.0% MV=000.0%"
            DB   "PV=000.0% PI   REV"
DEMO_PID_R:  DB   "SP=000.0% MV=000.0%"
            DB   "PV=000.0% PID  REV"
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
DEMO_PD_R:    DB    "SP=000.0% MV=000.0%"
              DB    "PV=000.0% PD   REV"
DEMO_ONOF_R:  DB    "SP=000.0% MV=000.0%"
              DB    "PV=000.0% ONOFF REV"

;************ LOOK_SP SUB ************
LOOK_SP:
    MOV   R3,#02H      ;LOOP

LSP_0: JB    SP_BUFH.7,LSP_2
    MOV   A,R4
    MOV   DPTR,#TABLE0
LSP_3: MOVC  A,@A+DPTR
    CJNE  R3,#1,LSP_1
    MOV   SP_BUFP,A
    RET
LSP_2: MOV   A,R4
    MOV   DPTR,#TABLE1
    SJMP  LSP_3

LSP_1: MOV   SP_BUF,A
    INC   R4
    DJNZ  R3,LSP_0

;************ LOOK_MV SUB ************
LOOK_MV:
    MOV   R3,#02H      ;LOOP
    . . .
LMV_0: JB    MV_BUFH.7,LMV_2
    MOV   A,R4
    MOV   DPTR,#TABLE0
LMV_3: MOVC  A,@A+DPTR
    CJNE  R3,#1,LMV_1
    MOV   MV_BUFP,A
    RET
LMV_2: MOV   A,R4
    MOV   DPTR,#TABLE1
    SJMP  LMV_3

LMV_1: MOV   MV_BUF,A
    INC   R4
    DJNZ  R3,LMV_0

;************ LOOK_PV SUB ************
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
LOOK_PV:
    MOV   R3,#02H      ;LOOP

LPV_0: JB    PV_BUFH.7,LPV_2
    MOV   A,R4
    MOV   DPTR,#TABLE0
LPV_3: MOVC   A,@A+DPTR
    CJNE  R3,#1,LPV_1
    MOV   PV_BUFP,A
    RET
LPV_2: MOV   A,R4
    MOV   DPTR,#TABLE1
    SJMP  LPV_3

LPV_1: MOV   PV_BUF,A
    INC   R4
    DJNZ  R3,LPV_0

;************ LOOK_TI SUB *********
LOOK_TI:
    MOV   R3,#02H      ;LOOP

LTI_0: JB    TI_BUFH.7,LTI_2
    MOV   A,R4
    MOV   DPTR,#TBL_T0
LTI_3: MOVC   A,@A+DPTR
    CJNE  R3,#1,LTI_1
    MOV   TI_BUF1,A
    RET
LTI_2: MOV   A,R4
    MOV   DPTR,#TBL_T1
    SJMP  LTI_3
LTI_1: MOV   TI_BUF0,A
    INC   R4
    DJNZ  R3,LTI_0

;************ LOOK_TD SUB *********
LOOK_TD:
    MOV   R3,#02H      ;LOOP

LTD_0: JB    TD_BUFH.7,LTD_2
    MOV   A,R4
    MOV   DPTR,#TBL_T0
LTD_3: MOVC   A,@A+DPTR
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    CJNE  R3,#1,LTD_1
    MOV   TD_BUF1,A
    RET
LTD_2: MOV   A,R4
    MOV   DPTR,#TBL_T1
    SJMP  LTD_3
LTD_1: MOV   TD_BUF0,A
    INC   R4
    DJNZ  R3,LTD_0


;************ LOOK_HL SUB **********
LOOK_HL:
    MOV   R3,#02H      ;LOOP

LHL_0: JB    HL_BUFH.7,LHL_2
    MOV   A,R4
    MOV   DPTR,#TABLE0
LHL_3: MOVC  A,@A+DPTR
    CJNE  R3,#1,LHL_1
    MOV   HL_BUFP,A
    RET
LHL_2: MOV   A,R4
    MOV   DPTR,#TABLE1
    SJMP  LHL_3
LHL_1: MOV   HL_BUF,A
    INC   R4
    DJNZ  R3,LHL_0

;************ LOOK_LL SUB ***********
LOOK_LL:
    MOV   R3,#02H      ;LOOP

LLL_0: JB    LL_BUFH.7,LLL_2
    MOV   A,R4
    MOV   DPTR,#TABLE0
LLL_3: MOVC  A,@A+DPTR
    CJNE  R3,#1,LLL_1
    MOV   LL_BUFP,A
    RET
LLL_2: MOV   A,R4
    MOV   DPTR,#TABLE1
    SJMP  LLL_3
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
LLL_1: MOV   LL_BUF,A
       INC   R4
       DJNZ  R3,LLL_0


;************* TABLE0 FOR HEX TO DEC ************

TABLE0: DB    00H,00H     ;0
        DB    00H,04H     ;1
        DB    00H,08H     ;2
        DB    01H,02H     ;3
        DB    01H,06H     ;4
        DB    02H,00H     ;5
        DB    02H,03H     ;6
        DB    02H,07H     ;7
        DB    03H,01H     ;8
        DB    03H,05H     ;9
        DB    03H,09H     ;10 **********
        DB    04H,03H     ;1
        DB    04H,07H     ;2
        DB    05H,01H     ;3
        DB    05H,05H     ;4
        DB    05H,09H     ;5
        DB    06H,03H     ;6
        DB    06H,07H     ;7
        DB    07H,01H     ;8
        DB    07H,05H     ;9
        DB    07H,08H     ;20 **********
        DB    08H,02H     ;1
        DB    08H,06H     ;2
        DB    09H,00H     ;3
        DB    09H,04H     ;4
        DB    09H,08H     ;5
        DB    10H,02H     ;6
        DB    10H,06H     ;7
        DB    11H,00H     ;8
        DB    11H,04H     ;9
        DB    11H,08H     ;30 **********
        DB    12H,02H     ;1
        DB    12H,05H     ;2
        DB    12H,09H     ;3
        DB    13H,03H     ;4
        DB    13H,07H     ;5
        DB    14H,01H     ;6
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
DB    14H,05H     ;7
DB    14H,09H     ;8
DB    15H,03H     ;9
DB    15H,07H     ;40 **********
DB    16H,01H     ;1
DB    16H,05H     ;2
DB    16H,09H     ;3
DB    17H,03H     ;4
DB    17H,07H     ;5
DB    18H,00H     ;6
DB    18H,04H     ;7
DB    18H,08H     ;8
DB    19H,02H     ;9
DB    19H,06H     ;50 **********
DB    20H,00H     ;1
DB    20H,04H     ;2
DB    20H,08H     ;3
DB    21H,02H     ;4
DB    21H,06H     ;5
DB    22H,00H     ;6
DB    22H,04H     ;7
DB    22H,07H     ;8
DB    23H,01H     ;9
DB    23H,05H     ;60 **********
DB    23H,09H     ;1
DB    24H,03H     ;2
DB    24H,07H     ;3
DB    25H,01H     ;4
DB    25H,05H     ;5
DB    25H,09H     ;6
DB    26H,03H     ;7
DB    26H,07H     ;8
DB    27H,01H     ;9
DB    27H,05H     ;70 **********
DB    27H,09H     ;1
DB    28H,02H     ;2
DB    28H,06H     ;3
DB    29H,00H     ;4
DB    29H,04H     ;5
DB    29H,08H     ;6
DB    30H,02H     ;7
DB    30H,06H     ;8
DB    31H,00H     ;9
DB    31H,04H     ;80 **********
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
DB    31H,08H     ;1
DB    32H,02H     ;2
DB    32H,06H     ;3
DB    32H,09H     ;4
DB    33H,03H     ;5
DB    33H,07H     ;6
DB    34H,01H     ;7
DB    34H,05H     ;8
DB    34H,09H     ;9
DB    35H,03H     ;90 **********
DB    35H,07H     ;1
DB    36H,01H     ;2
DB    36H,05H     ;3
DB    36H,09H     ;4
DB    37H,03H     ;5
DB    37H,07H     ;6
DB    38H,00H     ;7
DB    38H,04H     ;8
DB    38H,08H     ;9
DB    39H,02H     ;100 *********
DB    39H,06H     ;1
DB    40H,00H     ;2
DB    40H,04H     ;3
DB    40H,08H     ;4
DB    41H,02H     ;5
DB    41H,06H     ;6
DB    42H,00H     ;7
DB    42H,04H     ;8
DB    42H,07H     ;9
DB    43H,01H     ;110 *********
DB    43H,05H     ;1
DB    43H,09H     ;2
DB    44H,03H     ;3
DB    44H,07H     ;4
DB    45H,01H     ;5
DB    45H,05H     ;6
DB    45H,09H     ;7
DB    46H,03H     ;8
DB    46H,07H     ;9
DB    47H,01H     ;120 *********
DB    47H,05H     ;1
DB    47H,08H     ;2
DB    48H,02H     ;3
DB    48H,06H     ;4
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        DB    49H,00H      ;5
        DB    49H,04H      ;6
        DB    49H,08H      ;7  127D=7FH=50%
TABLE1: DB    50H,02H      ;8
        DB    50H,06H      ;9
        DB    51H,00H      ;130 *********
        DB    51H,04H      ;1
        DB    51H,08H      ;2
        DB    52H,02H      ;3
        DB    52H,06H      ;4
        DB    53H,00H      ;5
        DB    53H,03H      ;6
        DB    53H,07H      ;7
        DB    54H,01H      ;8
        DB    54H,05H      ;9
        DB    54H,09H      ;140 *********
        DB    55H,03H      ;1
        DB    55H,07H      ;2
        DB    55H,09H      ;3
        DB    56H,05H      ;4
        DB    56H,09H      ;5
        DB    57H,03H      ;6
        DB    57H,06H      ;7
        DB    58H,00H      ;8
        DB    58H,04H      ;9
        DB    58H,08H      ;150 *********
        DB    59H,02H      ;1
        DB    59H,06H      ;2
        DB    60H,00H      ;3
        DB    60H,04H      ;4
        DB    60H,08H      ;5
        DB    61H,02H      ;6
        DB    61H,06H      ;7
        DB    62H,00H      ;8
        DB    62H,04H      ;9
        DB    62H,08H      ;160 *********
        DB    63H,01H      ;1
        DB    63H,05H      ;2
        DB    63H,09H      ;3
        DB    64H,03H      ;4
        DB    64H,07H      ;5
        DB    65H,01H      ;6
        DB    65H,05H      ;7
        DB    65H,09H      ;8
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
DB    66H,03H     ;9
DB    66H,07H     ;170 *********
DB    67H,01H     ;1
DB    67H,05H     ;2
DB    67H,08H     ;3
DB    68H,02H     ;4
DB    68H,06H     ;5
DB    69H,00H     ;6
DB    69H,04H     ;7
DB    69H,08H     ;8
DB    70H,02H     ;9
DB    70H,06H     ;180 *********
DB    71H,00H     ;1
DB    71H,04H     ;2
DB    71H,08H     ;3
DB    72H,02H     ;4
DB    72H,06H     ;5
DB    72H,09H     ;6
DB    73H,03H     ;7
DB    73H,07H     ;8
DB    74H,01H     ;9
DB    74H,05H     ;190 *********
DB    74H,09H     ;1
DB    75H,03H     ;2
DB    75H,07H     ;3
DB    76H,01H     ;4
DB    76H,05H     ;5
DB    76H,09H     ;6
DB    77H,03H     ;7
DB    77H,07H     ;8
DB    78H,01H     ;9
DB    78H,04H     ;200 *********
DB    78H,08H     ;1
DB    79H,02H     ;2
DB    79H,06H     ;3
DB    80H,00H     ;4
DB    80H,04H     ;5
DB    80H,08H     ;6
DB    81H,02H     ;7
DB    81H,06H     ;8
DB    82H,00H     ;9
DB    82H,04H     ;210 *********
DB    82H,08H     ;1
DB    83H,01H     ;2
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
DB    83H,05H     ;3
DB    83H,09H     ;4
DB    84H,03H     ;5
DB    84H,07H     ;6
DB    85H,01H     ;7
DB    85H,05H     ;8
DB    85H,09H     ;9
DB    86H,02H     ;220 *********
DB    86H,07H     ;1
DB    87H,01H     ;2
DB    87H,05H     ;3
DB    87H,09H     ;4
DB    88H,02H     ;5
DB    88H,06H     ;6
DB    89H,00H     ;7
DB    89H,04H     ;8
DB    89H,08H     ;9
DB    90H,02H     ;230 *********
DB    90H,06H     ;1
DB    91H,00H     ;2
DB    91H,04H     ;3
DB    91H,08H     ;4
DB    92H,02H     ;5
DB    92H,06H     ;6
DB    92H,09H     ;7
DB    93H,03H     ;8
DB    93H,07H     ;9
DB    94H,01H     ;240 *********
DB    94H,05H     ;1
DB    94H,09H     ;2
DB    95H,03H     ;3
DB    95H,07H     ;4
DB    96H,01H     ;5
DB    96H,05H     ;6
DB    96H,09H     ;7
DB    97H,03H     ;8
DB    97H,07H     ;9
DB    98H,00H     ;250 *********
DB    98H,04H     ;1
DB    98H,08H     ;2
DB    99H,02H     ;3
DB    99H,06H     ;4
DB    00H,00H     ;5  255D=0FFH=100%
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
;************* TABLE0 FOR HEX TO DEC ************

TBL_T0: DB    00H,00H      ;0
        DB    00H,06H      ;1
        DB    00H,11H      ;2
        DB    00H,17H      ;3
        DB    00H,22H      ;4
        DB    00H,28H      ;5
        DB    00H,34H      ;6
        DB    00H,39H      ;7
        DB    00H,45H      ;8
        DB    00H,51H      ;9
        DB    00H,56H      ;10 **********
        DB    00H,62H      ;1
        DB    00H,68H      ;2
        DB    00H,73H      ;3
        DB    00H,79H      ;4
        DB    00H,85H      ;5
        DB    00H,90H      ;6
        DB    00H,96H      ;7
        DB    01H,02H      ;8
        DB    01H,07H      ;9
        DB    01H,13H      ;20 **********
        DB    01H,18H      ;1
        DB    01H,24H      ;2
        DB    01H,30H      ;3
        DB    01H,35H      ;4
        DB    01H,41H      ;5
        DB    01H,47H      ;6
        DB    01H,52H      ;7
        DB    01H,58H      ;8
        DB    01H,63H      ;9
        DB    01H,70H      ;30 **********
        DB    01H,75H      ;1
        DB    01H,80H      ;2
        DB    01H,86H      ;3
        DB    01H,92H      ;4
        DB    01H,97H      ;5
        DB    02H,03H      ;6
        DB    02H,09H      ;7
        DB    02H,14H      ;8
        DB    02H,20H      ;9
        DB    02H,25H      ;40 **********
        DB    02H,31H      ;1
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
DB    02H,37H     ;2
DB    02H,42H     ;3
DB    02H,48H     ;4
DB    02H,54H     ;5
DB    02H,60H     ;6
DB    02H,65H     ;7
DB    02H,71H     ;8
DB    02H,76H     ;9
DB    02H,82H     ;50 **********
DB    02H,88H     ;1
DB    02H,93H     ;2
DB    03H,00H     ;3
DB    03H,04H     ;4
DB    03H,10H     ;5
DB    03H,16H     ;6
DB    03H,21H     ;7
DB    03H,27H     ;8
DB    03H,33H     ;9
DB    03H,38H     ;60 **********
DB    03H,44H     ;1
DB    03H,50H     ;2
DB    03H,55H     ;3
DB    03H,61H     ;4
DB    03H,67H     ;5
DB    03H,72H     ;6
DB    03H,78H     ;7
DB    03H,83H     ;8
DB    03H,89H     ;9
DB    03H,95H     ;70 **********
DB    04H,00H     ;1
DB    04H,06H     ;2
DB    04H,12H     ;3
DB    04H,17H     ;4
DB    04H,23H     ;5
DB    04H,28H     ;6
DB    04H,34H     ;7
DB    04H,40H     ;8
DB    04H,45H     ;9
DB    04H,51H     ;80 **********
DB    04H,56H     ;1
DB    04H,62H     ;2
DB    04H,68H     ;3
DB    04H,74H     ;4
DB    04H,79H     ;5
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        DB    04H,85H     ;6
        DB    04H,91H     ;7
        DB    04H,96H     ;8
        DB    05H,02H     ;9
        DB    05H,07H     ;90 **********
        DB    05H,13H     ;1
        DB    05H,19H     ;2
        DB    05H,24H     ;3
        DB    05H,30H     ;4
        DB    05H,36H     ;5
        DB    05H,41H     ;6
        DB    05H,47H     ;7
        DB    05H,54H     ;8
        DB    05H,58H     ;9
        DB    05H,64H     ;100 *********
        DB    05H,69H     ;1
        DB    05H,75H     ;2
        DB    05H,81H     ;3
        DB    05H,86H     ;4
        DB    05H,92H     ;5
        DB    05H,98H     ;6
        DB    06H,03H     ;7
        DB    06H,09H     ;8
        DB    06H,15H     ;9
        DB    06H,20H     ;110 *********
        DB    06H,26H     ;1
        DB    06H,31H     ;2
        DB    06H,37H     ;3
        DB    06H,43H     ;4
        DB    06H,48H     ;5
        DB    06H,54H     ;6
        DB    06H,60H     ;7
        DB    06H,65H     ;8
        DB    06H,71H     ;9
        DB    06H,77H     ;120 *********
        DB    06H,82H     ;1
        DB    06H,88H     ;2
        DB    06H,93H     ;3
        DB    06H,99H     ;4
        DB    07H,05H     ;5
        DB    07H,10H     ;6
        DB    07H,16H     ;7  127D=7FH=50%
TBL_T1: DB    07H,22H     ;8
        DB    07H,27H     ;9
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
DB    07H,33H     ;130 *********
DB    07H,38H     ;1
DB    07H,44H     ;2
DB    07H,50H     ;3
DB    07H,55H     ;4
DB    07H,61H     ;5
DB    07H,67H     ;6
DB    07H,72H     ;7
DB    07H,78H     ;8
DB    07H,84H     ;9
DB    07H,89H     ;140 *********
DB    07H,95H     ;1
DB    08H,01H     ;2
DB    08H,06H     ;3
DB    08H,12H     ;4
DB    08H,17H     ;5
DB    08H,23H     ;6
DB    08H,29H     ;7
DB    08H,34H     ;8
DB    08H,40H     ;9
DB    08H,46H     ;150 *********
DB    08H,51H     ;1
DB    08H,57H     ;2
DB    08H,63H     ;3
DB    08H,68H     ;4
DB    08H,74H     ;5
DB    08H,79H     ;6
DB    08H,85H     ;7
DB    08H,91H     ;8
DB    08H,96H     ;9
DB    09H,02H     ;160 *********
DB    09H,08H     ;1
DB    09H,13H     ;2
DB    09H,19H     ;3
DB    09H,25H     ;4
DB    09H,30H     ;5
DB    09H,36H     ;6
DB    09H,41H     ;7
DB    09H,47H     ;8
DB    09H,53H     ;9
DB    09H,58H     ;170 *********
DB    09H,64H     ;1
DB    09H,70H     ;2
DB    09H,75H     ;3
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
DB    09H,81H     ;4
DB    09H,87H     ;5
DB    09H,92H     ;6
DB    09H,98H     ;7
DB    10H,03H     ;8
DB    10H,09H     ;9
DB    10H,15H     ;180 *********
DB    10H,20H     ;1
DB    10H,26H     ;2
DB    10H,32H     ;3
DB    10H,37H     ;4
DB    10H,43H     ;5
DB    10H,49H     ;6
DB    10H,54H     ;7
DB    10H,60H     ;8
DB    10H,66H     ;9
DB    10H,71H     ;190 *********
DB    10H,77H     ;1
DB    10H,82H     ;2
DB    10H,88H     ;3
DB    10H,94H     ;4
DB    10H,99H     ;5
DB    11H,05H     ;6
DB    11H,11H     ;7
DB    11H,16H     ;8
DB    11H,22H     ;9
DB    11H,27H     ;200 *********
DB    11H,33H     ;1
DB    11H,39H     ;2
DB    11H,44H     ;3
DB    11H,50H     ;4
DB    11H,56H     ;5
DB    11H,61H     ;6
DB    11H,67H     ;7
DB    11H,73H     ;8
DB    11H,78H     ;9
DB    11H,84H     ;210 *********
DB    11H,90H     ;1
DB    11H,95H     ;2
DB    12H,01H     ;3
DB    12H,06H     ;4
DB    12H,12H     ;5
DB    12H,18H     ;6
DB    12H,23H     ;7
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    DB    12H,29H     ;8
    DB    12H,35H     ;9
    DB    12H,40H     ;220 *********
    DB    12H,46H     ;1
    DB    12H,52H     ;2
    DB    12H,57H     ;3
    DB    12H,63H     ;4
    DB    12H,68H     ;5
    DB    12H,74H     ;6
    DB    12H,80H     ;7
    DB    12H,85H     ;8
    DB    12H,91H     ;9
    DB    12H,97H     ;230 *********
    DB    13H,02H     ;1
    DB    13H,08H     ;2
    DB    13H,14H     ;3
    DB    13H,19H     ;4
    DB    13H,25H     ;5
    DB    13H,30H     ;6
    DB    13H,36H     ;7
    DB    13H,42H     ;8
    DB    13H,47H     ;9
    DB    13H,53H     ;240 *********
    DB    13H,59H     ;1
    DB    13H,64H     ;2
    DB    13H,70H     ;3
    DB    13H,76H     ;4
    DB    13H,81H     ;5
    DB    13H,87H     ;6
    DB    13H,92H     ;7
    DB    13H,98H     ;8
    DB    14H,04H     ;9
    DB    14H,09H     ;250 *********
    DB    14H,15H     ;1
    DB    14H,21H     ;2
    DB    14H,26H     ;3
    DB    14H,32H     ;4
    DB    14H,38H     ;5  255D=0FFH=100%

;********* DELAY SUB **********;;DELAY FOR TITLE AND LOGO
DELAY: MOV   R7,#7FH
DY1:   MOV   R6,#0
DY2:   MOV   R5,#10H
       DJNZ  R5,$
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    DJNZ  R6,DY2
    DJNZ  R7,DY1
    RET

;********* DELAY_Q SUB **********;;DELAY FOR INC AND DEC QUICK
DELAY_Q:
    MOV   R7,#18H
DY5:  MOV   R6,#0
    DJNZ  R6,$
    DJNZ  R7,DY5
    RET

;********* DELAY_SL SUB **********;;DELAY FOR INC AND DEC SLOW
DELAY_SL:    MOV   R5,#1
DY9:  MOV   R7,#0
DY8:  MOV   R6,#0
    DJNZ  R6,$
    DJNZ  R7,DY8
    DJNZ  R5,DY9
    RET

;********* DELAY2 SUB **********;;DELAY FOR KP AND KEYPRESS
DELAY2: MOV   R7,#2
DY3:  MOV   R6,#0
DY4:  MOV   R5,#0
    DJNZ  R5,$
    DJNZ  R6,DY4
    DJNZ  R7,DY3
    RET

;*********INPUT  SUB***********;
INPUT: LCALL  INPUT1
    LCALL  DELAY2
    LCALL  INPUT1
    RET

;********* INPUT1 SUB **********
INPUT1: MOV   DPTR,#PORTC
    MOVX  A,@DPTR
    JZ    A_50
    RET
A_50:  MOV   A,#0FFH
    RET
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
;******* SUB STEP_RX ************
STEP_RX:
    MOV   A,RUN_STEP
    LCALL  RUN_T1
    RET

RUN_T1: LCALL  WR_COMM
RUN_T2: CLR   A
    MOV   A,@R1
    LCALL  WRITE
    RET

;******** SUB RUN_2C *************
RUN_2C: MOV   A,RUN_STEP    ;L1=8A,L2CA
    LCALL  RUN_2C1
    RET
RUN_2C1:LCALL  WR_COMM
    MOV   R2,#2        ;2 CHAR
RUN_2C2:CLR   A
    MOV   A,@R1
    LCALL  WRITE
    INC   R1
    DJNZ  R2,RUN_2C2
    RET

;******** SUB RUN_2R1 *************
RUN_2R1:
    MOV   A,RUN_STEP
    LCALL  RUN_2C1
    RET

;******** SUB RUN_2R2 *************
RUN_2R2:
    MOV   A,RUN_STEP
    LCALL  RUN_2C1
    RET
;******** SUB RUN_2D *************
RUN_2D: MOV   A,RUN_STEP    ;L1=8A,L2CA
    LCALL  RUN_2D1
    RET
RUN_2D1:LCALL  WR_COMM
    MOV   A,@R1
    LCALL  WRITE
    RET
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
;******** SUB RUN_2R11 ************
RUN_2R11:
    MOV   A,RUN_STEP
    LCALL  RUN_2D1
    RET

;******** SUB RUN_2R21************
RUN_2R21:
    MOV   A,RUN_STEP
    LCALL  RUN_2D1
    RET

;******** HTOA SUB ***********
;CONVERT HEX TO ASCII
;IN  = A
;OUT = R2,R3
;REG = A,R2,R3

HTOA:  PUSH   ACC
    SWAP   A
    LCALL  HTOAS
    MOV   R2,A
    POP   ACC
    LCALL  HTOAS
    MOV   R3,A
    RET
HTOAS:  ANL   A,#0FH
    CJNE   A,#0AH,$+3
    JNC   HTOAS1
    ORL   A,#30H
    RET
HTOAS1: SUBB   A,#9
    ORL   A,#40H
    RET

;******** HTOA1 SUB ***********
;CONVERT HEX TO ASCII
;IN  = A
;OUT = A
HTOA1:  ANL   A,#0FH
    CJNE   A,#0AH,$+3
    JNC   HTOA12
    ORL   A,#30H
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    RET
HTOA12: SUBB   A,#9
    ORL    A,#40H
    RET

;******** SUB WR_COMM ********
WR_COMM:
    PUSH   DPH
    PUSH   DPL
    MOV    DPTR,#COM_PORT
    MOVX   @DPTR,A
    MOV    DPTR,#RD_BUSY
BUSYC: MOVX   A,@DPTR
    JB     ACC.7,BUSYC
    POP    DPL
    POP    DPH
    RET

;******** SUB WRITE **********
WRITE: PUSH   DPL
    PUSH   DPH
    MOV    DPTR,#DAT_PORT
    MOVX   @DPTR,A
    MOV    DPTR,#RD_BUSY
BUSY1: MOVX   A,@DPTR
    JB     ACC.7,BUSY1
    POP    DPH
    POP    DPL
    RET

;******** SUB DISPLAY ********
DISPLAY:
    MOV    A,#80H       ;SET ADD LINE 1
    LCALL  LCD_RUN
    MOV    A,#0C0H      ;SET ADD LINE 2
    LCALL  LCD_RUN
    RET
LCD_RUN:
    LCALL  WR_COMM
    MOV    R2,#20       ;20 CHAR
LCD_RUN1:
    CLR    A
    MOVC   A,@A+DPTR
    LCALL  WRITE
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    INC   DPTR
    DJNZ  R2,LCD_RUN1
    RET

;********** DIS_MV SUB *********
DIS_MV:
    MOV   A,MV_BUF     ;***Take NEW VALUE to display***
    LCALL  HTOA
    MOV   BUF_LCD,R2
    MOV   BUF_LCD+1,R3
    MOV   RUN_STEP,#8FH
    MOV   R1,#BUF_LCD
    LCALL  RUN_2R1     ;*******************************

    MOV   A,MV_BUFP    ;***Take NEW VALUE to display***
    LCALL  HTOA
    MOV   R2,#0
    MOV   BUF_LCD+2,R3
    MOV   RUN_STEP,#92H
    MOV   R1,#BUF_LCD+2
    LCALL  RUN_2R11    ;*******************************

    MOV   A,MV_BUFH
    CJNE  A,#0FFH,A_53
    MOV   R3,#31H      ;=1 (100.0)
    MOV   R2,#0
    MOV   BUF_LCD+3,R3
    MOV   RUN_STEP,#8EH
    MOV   R1,#BUF_LCD+3
    LCALL  RUN_2R11    ;*******************************
    SJMP  A_103

A_53:  MOV   R3,#30H     ;=0 (0XX.X)
    MOV   R2,#0
    MOV   BUF_LCD+3,R3
    MOV   RUN_STEP,#8EH
    MOV   R1,#BUF_LCD+3
    LCALL  RUN_2R11    ;*******************************

A_103: MOV   A,MV_BUFH
    CLR   C
    RLC   A
    MOV   R4,A
    LCALL  LOOK_MV
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    RET

;********** DIS_PV SUB *********
DIS_PV:
    MOV   A,PV_BUF     ;***Take NEW VALUE to display***
    LCALL  HTOA
    MOV   BUF_LCD,R2
    MOV   BUF_LCD+1,R3
    MOV   RUN_STEP,#0C4H
    MOV   R1,#BUF_LCD
    LCALL  RUN_2R1     ;*******************************

    MOV   A,PV_BUFP    ;***Take NEW VALUE to display***
    LCALL  HTOA
    MOV   R2,#0
    MOV   BUF_LCD+2,R3
    MOV   RUN_STEP,#0C7H
    MOV   R1,#BUF_LCD+2
    LCALL  RUN_2R11    ;*******************************

    MOV   A,PV_BUFH
    CJNE  A,#0FFH,DPV_0
    MOV   R3,#31H      ;=1 (100.0)
    MOV   R2,#0
    MOV   BUF_LCD+3,R3
    MOV   RUN_STEP,#0C3H
    MOV   R1,#BUF_LCD+3
    LCALL  RUN_2R11    ;*******************************
    SJMP  DPV_1

DPV_0: MOV   R3,#30H     ;=0 (0XX.X)
    MOV   R2,#0
    MOV   BUF_LCD+3,R3
    MOV   RUN_STEP,#0C3H
    MOV   R1,#BUF_LCD+3
    LCALL  RUN_2R11    ;*******************************

DPV_1: MOV   A,PV_BUFH
    CLR   C
    RLC   A
    MOV   R4,A
    LCALL  LOOK_PV
    RET
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
;********** MAIN_CONTROL ***********
MAIN_CON:
    JB    21H.0,A_112   ;MAN
    JB    21H.1,A_100   ;P
    JB    21H.2,A_114   ;PI
    JB    21H.3,A_115   ;PID
    JB    21H.4,A_116   ;PD
    JB    21H.5,A_109   ;ON-OFF

    LJMP  DIS_SP        ;NO KEY PRESS

A_112: LJMP  MAN_ACT
A_100: LJMP  P_ACT
A_114: LJMP  P_ACT
A_115: LJMP  P_ACT
A_116: LJMP  P_ACT
A_109: LJMP  ONOFF_ACT

MAIN_PID:
    LCALL  DIS_MV
    MOV    A,MV_BUFH
    MOV    DPTR,#PORTB
    MOVX   @DPTR,A

    LJMP   DIS_SP

;********** M_BIAS SUB **********
M_BIAS: CLR   C
    JB    21H.6,MB_0
    MOV   A,R5
    ADD   A,#7FH        ;M(BIAS)+(P)
    MOV   R7,A
    JC    MB_1          ;IF OVER 0FFH, MV_BUFH=0FFH
    SJMP  MB_2
MB_1:  MOV   R7,#0FFH
MB_2:  MOV   MV_BUFH,R7
    SJMP  MB_5

MB_0:  MOV   A,#7FH
    SUBB  A,R5          ;M(BIAS)+(-P)
    JC    MB_3          ;IF LESS THAN #0H,MV_BUFH=#0 ONLY
    SJMP  MB_4
MB_3:  MOV   A,#0
    CLR   C
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
MB_4:  JZ    JK_200
       DEC   A          ;FOR P_ACT
JK_200: MOV  MV_BUFH,A
MB_5:  LJMP  MAIN_PID

;********* P_ACTION SUB **********
;      MV = KP(SP-PV)      ;M(BIAS) ADD AFTER

P_ACT: JB    20H.6,P_0     ;DIRECT
       JNB   20H.6,P_1     ;REVERSE

P_0:   MOV   A,SP_BUFH
       MOV   R3,PV_BUFH
       SUBB  A,R3          ;SP-PV
       MOV   R4,A          ;R4=SP-PV
       SJMP  P_2

P_1:   MOV   A,PV_BUFH
       MOV   R3,SP_BUFH
       SUBB  A,R3          ;PV-SV
       MOV   R4,A          ;R4=PV-SP

P_2:   JC    P_3
       MOV   B,R4
       MOV   A,KP_BUFH
       MUL   AB            ;KP(SP-PV)
       MOV   R4,B
       MOV   B,#0
       CJNE  R4,#0H,P_4    ;IF OVER #0FFH MOV R5,#0FFH
       MOV   R5,A          ;R5=SENDING TO M_BIAS
       MOV   R7,A          ;R7=R5=SENDING
       CLR   21H.6         ;+
       LJMP  P_6
P_4:   MOV   R5,#0FFH      ;R5=SENDING TO M_BIAS
       MOV   R7,#0FFH      ;R7=R5=SENDING
       CLR   21H.6
       LJMP  P_6

P_3:   MOV   A,#0FFH
       SUBB  A,R4
       INC   A
       MOV   R4,A          ;R4=-(SP-PV)
       MOV   B,R4
       MOV   A,KP_BUFH
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    MUL   AB          ;-KP(SP-PV)
    MOV   R4,B
    MOV   B,#0
    CJNE  R4,#0H,P_5
    MOV   R5,A        ;R5=SENDING TO M_BIAS
    MOV   R7,A        ;R7=R5=SENDING
    SETB  21H.6       ;-
    SJMP  P_6
P_5: MOV  R5,#0FFH    ;R5=SENDING TO M_BIAS
    MOV   R7,#0FFH    ;R7=R5=SENDING
    SETB  21H.6
P_6: JB   21H.2,I_ACT ;SELECT I? PI
    JB    21H.3,I_ACT ;SELECT I? PID
    JB    21H.4,P_7   ;SELECT D? PD
    LJMP  M_BIAS
P_7: LJMP  D_ACT

;********* I_ACTION SUB **********
;    MV = KP(SP-PV)*DELTA(Ti)/Ti   ;M(BIAS) ADD AFTER

I_ACT: JB   20H.6,I_0    ;DIRECT
    JNB   20H.6,I_1   ;REVERSE

I_0: CLR   C
    MOV   A,SP_BUFH
    MOV   R3,PV_BUFH
    SUBB  A,R3        ;SP-PV
    MOV   R4,A        ;R4=SP-PV
    SJMP  I_2

I_1: CLR   C
    MOV   A,PV_BUFH
    MOV   R3,SP_BUFH
    SUBB  A,R3        ;PV-SV
    MOV   R4,A        ;R4=PV-SP
I_2: JC    I_3
    SJMP  I_4
I_3: MOV   A,#0FFH
    SUBB  A,R4
    INC   A
    MOV   R4,A        ;R4=-(SP-PV)
I_4: MOV   B,KP_BUFH
    MUL   AB
    MOV   R4,B
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    MOV   B,#0
    CJNE  R4,#0H,I_5    ;IF OVER #0FFH MOV R5,#0FFH
    MOV   R5,A          ;R5=SENDING TO M_BIAS
    SJMP  I_6
I_5: MOV  R5,#0FFH      ;R5=SENDING TO M_BIAS
I_6: JB   21H.7,I_7
    MOV   DTI_BUFH,#1   ;START DTI_BUFH,#1
I_7: MOV  A,R5
    MOV   B,DTI_BUFH
    MUL   AB
    MOV   R2,A
    MOV   R3,B
    MOV   R6,#1
I_8: CLR  C
    MOV   A,R6
    MOV   B,TI_BUFH
    MUL   AB
    MOV   R4,A
    MOV   R5,B
    MOV   A,R5
    SUBB  A,R3
    JC    I_9
    SJMP  I_10
I_9: INC  R6
    LJMP  I_8
I_10: CLR  C
    MOV   A,DTI_BUFH
    SUBB  A,TI_BUFH
    JC    I_11
    CLR   21H.7
    SJMP  I_12
I_11: INC  DTI_BUFH
    SETB  21H.7
    SJMP  I_12
I_12: CLR  C
    MOV   A,R6
    ADD   A,R7          ;R7=R5 FROM P_ACT
    JC    I_13
    MOV   R5,A          ;R5=SENDING TO M_BIAS
    SJMP  I_14
I_13: MOV  R5,#0FFH     ;R5=SENDING TO M_BIAS
I_14: JB   21H.3,D_ACT
    LJMP  M_BIAS
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
;********** D_ACT SUB **************
;    MV=KP(En-En-1)TD/Delta TD

D_ACT: JB    20H.7,D0
    MOV   DTD_BUFH,#1
D0:  MOV   A,TD_BUFH
    MOV   B,DTD_BUFH
    DIV   AB
    MOV   TD_DTD,A
    MOV   A,SP_BUFH
    MOV   R2,PV_BUFH
    CLR   C
    SUBB  A,R2
    JC    D1
    CLR   22H.1
    JB    22H.2,D2
    CLR   C
    MOV   EN_BUFH,A
    SUBB  A,EN_1_BUFH
D11: JNC   D3
    SETB  22H.0
    LJMP  D7

D2:  CLR   C
    MOV   A,EN_1_BUFH
    JC    D8
    CLR   22H.0
    LJMP  D9
D8:  MOV   A,#0FFH
    CLR   22H.0
    LJMP  D9
D1:  SETB  22H.1
    JB    22H.2,D10
    CLR   C
    MOV   EN_BUFH,A
    ADD   A,EN_1_BUFH
    JC    D24
    SETB  22H.0
    LJMP  D9
D24: MOV   A,#0FFH
    SETB  22H.0
    LJMP  D9
D10: MOV   A,EN_1_BUFH
    SUBB  A,EN_BUFH
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    LJMP  D11

D3:  CLR   22H.0
D7:  MOV   R3,A
     MOV   A,#0FFH
     SUBB  A,R3
     INC   A
D9:  MOV   B,TD_DTD
     MUL   AB
     MOV   R3,A
     MOV   R7,B
     MOV   A,R7
     JNZ   D4
     LJMP  D6
D4:  MOV   R3,#0FFH
D6:  MOV   A,R3
     MOV   B,KP_BUFH
     MUL   AB
     MOV   R6,A
     MOV   R7,B
     MOV   A,R7
     JNZ   D5
     LJMP  D12
D5:  MOV   R6,#0FFH
D12: JB    20H.6,D14
     JNB   20H.6,D13
D13: CPL   22H.0
D14: JB    21H.6,D15
     JB    22H.0,D16
     CLR   C
     MOV   A,R6
     ADD   A,R5
     JC    D17
     MOV   R5,A
     CLR C
     LJMP  END
D17: MOV   R5,#0FFH
     CLR   21H.6
     LJMP  END
D16: CLR   C
     MOV   A,R5
     SUBB  A,R6
D21: JC    D18
     MOV   R5,A
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        CLR   21H.6
        LJMP  END
D18:    MOV   R5,A
        MOV   A,#0FFH
        SUBB  A,R5
        INC   A
        MOV   R5,A
        SETB  21H.6
        LJMP  END
D15:    JB    22H.0,D19
        MOV   A,R6
        SUBB  A,R5
        LJMP  D21
D19:    CLR   C
        MOV   A,R6
        ADD   A,R5
        JC    D20
        MOV   R5,A
        SETB  21H.6
        LJMP  END
D20:    MOV   R5,#0FFH
        SETB  21H.6
        LJMP  END

END:    MOV   A,DTD_BUFH
        CJNE  A,TD_BUFH,D25
        CLR   20H.7
        LJMP  D22
D25:    SETB  20H.7
        INC   DTD_BUFH

D22:    JB    22H.1,D23
        MOV   EN_1_BUFH,EN_BUFH
        CLR   22H.2
        LJMP  M_BIAS
D23:    MOV   EN_1_BUFH,EN_BUFH
        SETB  22H.2
        LJMP  M_BIAS

;********** ONOFF_ACT SUB **********
;MV=100% IF SP>PV
;MV=000% IF SP<PV

ONOFF_ACT:
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    JB    20H.6,O_0     ;DIRECT
    JNB   20H.6,O_1     ;REVERSE

O_0:  CLR   C
    MOV   A,PV_BUFH
    SUBB  A,SP_BUFH     ;C=1;PV<SP   C=0;PV>SP
    SJMP  O_2

O_1:  CLR   C
    MOV   A,SP_BUFH
    SUBB  A,PV_BUFH     ;C=1;PV>SP   C=0;PV<SP

O_2:  JC    O_3
    MOV   MV_BUF,#0
    MOV   MV_BUFP,#0
    MOV   MV_BUFH,#0
    LJMP  O_4

O_3:  MOV   MV_BUF,#0
    MOV   MV_BUFP,#0
    MOV   MV_BUFH,#0FFH

O_4:  LCALL DIS_MV
    MOV   A,MV_BUFH
    MOV   DPTR,#PORTB
    MOVX  @DPTR,A

    LJMP  DIS_SP

;************ MAN_ACT SUB *************
;INPUT =SP_BUFH
;OUTPUT=MV_BUFFH
MAN_ACT:
    JB    20H.6,M_0     ;DIRECT
    JNB   20H.6,M_1     ;REVERSE

M_0:  MOV   MV_BUFH,SP_BUFH
    SJMP  M_2

M_1:  MOV   A,#0FFH
    MOV   R3,SP_BUFH
    SUBB  A,R3
    MOV   MV_BUFH,A
M_2:  LCALL DIS_MV
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        MOV   A,MV_BUFH
        MOV   DPTR,#PORTB
        MOVX  @DPTR,A

        LJMP  DIS_SP

    END
```


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
; FILENAME    N03.ASM ( ST.No 03 )
; DESCRIPTION  PROGRAM FOR Small Scale DCS (Analog INPUT)
; HARDWARE    ANT-32
; ASSEMBLER   SXA51
; START-DATE  28/08/96
; SOFTWARE ENG  KMITL Power User
; COMPANY     KMITL , KMITNB


;********** Dim Variable ***********

V0    EQU   30H
V1    EQU   31H
V2    EQU   32H
V3    EQU   33H
V4    EQU   34H
V5    EQU   35H
V6    EQU   36H
V7    EQU   37H

V_BlockST    EQU   40H
V_Block00    EQU   41H
V_Block01    EQU   42H
V_Block02    EQU   43H
V_Block03    EQU   44H
V_Block04    EQU   45H
V_Block05    EQU   46H
V_Block06    EQU   47H
V_Block07    EQU   48H
V_BlockEND   EQU   49H

;*****************************;
;    SERIAL USE INT
;     ANT-32,REM31
;*****************************;
    ORG   0000H
    AJMP  ST

    ORG   0023H  ;REM31 INT Vector RI+TI : MCS-51(0023H)
    JBC   RI,RECV                    ;TI+RI Vector Action
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        LJMP   OUT    ;if TI reti OUT

RECV:
        LCALL  RxBlock ;Save MV to V_Block
        mov    a,V_BlockST
        cjne   a,#21H ,OUT    ;!
        mov    a,V_Block00
        cjne   a,#30H ,OUT    ;0
        mov    a,V_Block01
        cjne   a,#33H ,OUT    ;3


        lcall  send
        JNB    TI,$
        CLR    TI
        SETB   ES

OUT:    RETI

SEND:
        setb   p3.4
        LCALL  PVIB    ;IN PV,Save to V_Block
        LCALL  CHKSUM
        MOV    V_Block07,A
        LCALL  TxBlock ;Tx V_Block
        clr    p3.4
        RET

        ORG    0100H
;****************** Set Const ****************
ST:         LCALL  INIv
           clr    p3.4
;SetIntV:      mov    dptr,#8008h             ;Not use for EPROM
;           mov    a,#80h                  ;
;           movx   @dptr,a                 ;
;           inc    dptr                    ;
;           mov    a,#23h                  ;
;           movx   @dptr,a                 ;

OVER:
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    MOV   SCON,#50H                ;Serial Mode 1 9600
    ;MOV   PCON,#10000000B             ;x2 = 19200
    MOV   TMOD,#20H                ;
    MOV   TH1,#0FDH                ;
    SETB  TR1                      ;
    SETB  EA                       ;
    SETB  ES                       ;

                                   ;
        SJMP  $ ;add1              ; STOP

INIv:
    MOV   V_BlockST,#00100001B  ;!
    MOV   V_BlockEND,#00100011B  ;#

    MOV   V_Block00,#30H       ;0   ST.No.
    MOV   V_Block01,#33H       ;3   ST.No.
    MOV   V_Block02,#57H       ;W   General Command & Status
    MOV   V_Block03,#2BH       ;+   DATA H
    MOV   V_Block04,#2BH       ;+   DATA L
    MOV   V_Block05,#2BH       ;+   MultiPurpose
    MOV   V_Block06,#2BH       ;+   MultiPurpose
    MOV   V_Block07,#24H       ;$   ChkSUM

    RET

;MAIN1:



;************** TxBlock ************          ;Tx V_Block

                               ;
TxBlock:    MOV   R0,#40H              ;
        LCALL  mDELAY ;RS-485
TxBLoop:    MOV   A,@R0                ;
        LCALL  Tx1Byte             ;
        INC   R0                   ;
        CJNE  R0,#4AH,TxBLoop      ;
        RET                        ;
                               ;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                                    ;
; ********** Tx1Byte SUB **********                ;
; SEND 1 BYTE                                ;
; IN  = A                                    ;
; REG = NO                                   ;
                                    ;
Tx1Byte:    CLR   TI                         ;
            CLR   ES                         ;
            MOV   SBUF,A                     ;
            JNB   TI,$                       ;
            RET                              ;

;************** RxBlock ************
RxBlock:    LCALL  Rx1Byte
            ;MOV   V0,A
            mov  V_BlockST,A    ;SAVE 1ST from PC TO BLOCK
            cjne  a,#21H,RxOUT  ;!
        LCALL  Rx1Byte          ;ST.No. Byte 1
            ;MOV   V0,A
            mov  V_Block00,A    ;SAVE from PC TO BLOCK
            cjne  a,#30H,RxOUT  ;0
        LCALL  Rx1Byte          ;ST.No. Byte 2
            ;MOV   V0,A
            mov  V_Block01,A    ;SAVE from PC TO BLOCK
            cjne  a,#33H,RxOUT  ;3
        LCALL  Rx1Byte          ;GSC Byte
            ;MOV   V0,A
        LCALL  Rx1Byte
            ;MOV   V0,A
            mov  V_Block03,A    ;SAVE MV from PC TO BLOCK
        LCALL  Rx1Byte
            ;MOV   V0,A
            mov  V_Block04,A    ;SAVE MV from PC TO BLOCK
        LCALL  Rx1Byte      ;+
            ;MOV   V0,A
        LCALL  Rx1Byte      ;+
            ;MOV   V0,A
        LCALL  Rx1Byte      ;ChkSum
            ;MOV   V0,A
RxOUT:      RET
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
; ********** Rx1Byte SUB **********
Rx1Byte:    MOV   A,SBUF
            JNB   RI,$          ;WAIT FOR RECEIVE OK
            CLR   RI
            RET

;**********PV IN & to V_Block SUB**********

PVIB:

            mov   a,p1                      ;Get input
            MOV   V1,A  ;SAVE INPUT TO V1           ;Save to
                                    ;V_Block
            MOV   A,V1                      ;
            LCALL  HTOA                     ;
             mov  V_Block03,r2  ;SAVE PV TO V_BLOCK    ;
             mov  V_Block04,r3  ;SAVE PV TO V_BLOCK    ;

                    RET


;******** HTOA SUB **********
;CONVERT HEX TO ASCII
;IN  = A
;OUT = R2,R3
;REG = A,R2,R3
HTOA:   PUSH  ACC
            SWAP  A
            LCALL HTOAS
            MOV   R2,A
            POP   ACC
            LCALL HTOAS
            MOV   R3,A
            RET
HTOAS:  ANL   A,#0FH
            CJNE  A,#0AH,$+3
            JNC   HTOAS1
            ORL   A,#30H
            RET
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
HTOAS1:  SUBB A,#9
         ORL  A,#40H
         RET


; ********** ATOH SUB **********
; ASCII TO HEX CONVERT
; IN  = R2,R3  30H,41H
; OUT = A      0AH
; REG = A,R2

ATOH:  MOV   A,R2
       LCALL ATOHS
       SWAP  A
       MOV   R2,A
       MOV   A,R3
       LCALL ATOHS
       ORL   A,R2
       RET
ATOHS: CJNE  A,#'A',$+3
       JC    ATOHS1
       ADD   A,#9
ATOHS1: ANL  A,#0FH
       RET


; ********** CHKSUM SUB **********
;Sum V_Block00-06 without carry
;IN = V_Block00-06
;OUT = A

CHKSUM:
         MOV   A,#00H
         ADD   A,V_Block00
         ADD   A,V_Block01
         ADD   A,V_Block02
         ADD   A,V_Block03
         ADD   A,V_Block04
         ADD   A,V_Block05
         ADD   A,V_Block06
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        RET
; ********** DELAY SUB **********
; DELAY SUBROUTINE
; IN = R2
; REG = R2,R3,R4

DELAY:      MOV   R3,#0
DELAY1:     MOV   R4,#0
                  DJNZ  R4,$
                  DJNZ  R3,DELAY1
                  DJNZ  R2,DELAY
                  RET
; ********** mDELAY SUB **********
; mDELAY SUBROUTINE
; REG = R3,R4

mDELAY:     MOV   R3,#3FH
mDELAY1:    MOV   R4,#0
                  DJNZ  R4,$
        DJNZ  R3,mDELAY1
                  RET


;-------------------------------
;---------------------------------------
;------------------- END ---------------------
;---------------------------------------
;-------------------------------
END
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    MOV   SCON,#50H              ;Serial Mode 1 9600
    ;MOV   PCON,#10000000B       ;x2 = 19200
    MOV   TMOD,#20H              ;
    MOV   TH1,#0FDH              ;
    SETB  TR1                    ;
    SETB  EA                     ;
    SETB  ES                     ;

                                 ;
        SJMP  $ ;add1            ; STOP

INIv:
    MOV   V_BlockST,#00100001B  ;!
    MOV   V_BlockEND,#00100011B ;#

    MOV   V_Block00,#30H     ;0   ST.No.
    MOV   V_Block01,#34H     ;4   ST.No.
    MOV   V_Block02,#57H     ;W   General Command & Status
    MOV   V_Block03,#2BH     ;+   DATA H
    MOV   V_Block04,#2BH     ;+   DATA L
    MOV   V_Block05,#2BH     ;+   MultiPurpose
    MOV   V_Block06,#2BH     ;+   MultiPurpose
    MOV   V_Block07,#24H     ;$   ChkSUM

    RET

;MAIN1:



;************** TxBlock ************          ;Tx V_Block
                                 ;
TxBlock:    MOV   R0,#40H        ;
        LCALL  mDELAY ;RS-485
TxBLoop:    MOV   A,@R0          ;
        LCALL  TxIByte           ;
        INC   R0                 ;
        CJNE  R0,#4AH,TxBLoop    ;
        RET                      ;
                                 ;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                                ;
; ********** Tx1Byte SUB **********          ;
; SEND 1 BYTE                          ;
; IN = A                             ;
; REG = NO                             ;
                                ;
Tx1Byte:    CLR   TI                 ;
            CLR   ES                 ;
            MOV   SBUF,A             ;
            JNB   TI,$               ;
            RET                      ;

;************** RxBlock ************
RxBlock:    LCALL  Rx1Byte
            ;MOV   V0,A
            mov  V_BlockST,A    ;SAVE 1ST from PC TO BLOCK
            cjne  a,#21H,RxOUT  ;!
        LCALL  Rx1Byte          ;ST.No. Byte 1
            ;MOV   V0,A
            mov  V_Block00,A    ;SAVE from PC TO BLOCK
            cjne  a,#30H,RxOUT  ;0
        LCALL  Rx1Byte          ;ST.No. Byte 2
            ;MOV   V0,A
            mov  V_Block01,A    ;SAVE from PC TO BLOCK
            cjne  a,#34H,RxOUT  ;4
        LCALL  Rx1Byte          ;GSC Byte
            ;MOV   V0,A
        LCALL  Rx1Byte
            ;MOV   V0,A
            mov  V_Block03,A    ;SAVE MV from PC TO BLOCK
        LCALL  Rx1Byte
            ;MOV   V0,A
            mov  V_Block04,A    ;SAVE MV from PC TO BLOCK
        LCALL  Rx1Byte      ;+
            ;MOV   V0,A
        LCALL  Rx1Byte      ;+
            ;MOV   V0,A
        LCALL  Rx1Byte      ;ChkSum
            ;MOV   V0,A
RxOUT:      RET
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
; ********** Rx1Byte SUB **********
Rx1Byte:    MOV   A,SBUF
            JNB   RI,$          ;WAIT FOR RECEIVE OK
            CLR   RI
            RET



;**********MV OUT SUB**********

MVO:

            mov   r2,v_block03          ;MV from V_Block
            mov   r3,v_block04          ;
            lcall atoh  ;R2,R3 -> A     ;Convert to HEX
            mov   v2,a                  ;Save to V2

      MOV   A,V2                        ;OUT MV to Port1
            MOV   p1,A              ;


                  RET

;******** HTOA SUB ***********
;CONVERT HEX TO ASCII
;IN = A
;OUT = R2,R3
;REG = A,R2,R3
HTOA:   PUSH  ACC
            SWAP  A
            LCALL HTOAS
            MOV   R2,A
            POP   ACC
            LCALL HTOAS
            MOV   R3,A
            RET
HTOAS:  ANL   A,#0FH
            CJNE  A,#0AH,$+3
            JNC   HTOAS1
            ORL   A,#30H
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            RET
HTOAS1:  SUBB A,#9
            ORL  A,#40H
            RET


; ********** ATOH SUB **********
; ASCII TO HEX CONVERT
; IN  = R2,R3  30H,41H
; OUT = A      0AH
; REG = A,R2

ATOH:   MOV   A,R2
     LCALL ATOHS
     SWAP  A
     MOV   R2,A
     MOV   A,R3
     LCALL ATOHS
     ORL   A,R2
     RET
ATOHS:  CJNE  A,#'A',$+3
     JC    ATOHS1
     ADD   A,#9
ATOHS1: ANL   A,#0FH
     RET


; ********** CHKSUM SUB **********
;Sum V_Block00-06 without carry
;IN = V_Block00-06
;OUT = A

CHKSUM:
          MOV    A,#00H
          ADD    A,V_Block00
          ADD    A,V_Block01
          ADD    A,V_Block02
          ADD    A,V_Block03
          ADD    A,V_Block04
          ADD    A,V_Block05
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        ADD   A,V_Block06

        RET
; ********** DELAY SUB **********
; DELAY SUBROUTINE
; IN = R2
; REG = R2,R3,R4

DELAY:      MOV   R3,#0
DELAY1:     MOV   R4,#0
            DJNZ  R4,$
            DJNZ  R3,DELAY1
            DJNZ  R2,DELAY
            RET

; ********** mDELAY SUB **********
; mDELAY SUBROUTINE
; REG = R3,R4

mDELAY:     MOV   R3,#3FH
mDELAY1:    MOV   R4,#0
            DJNZ  R4,$
        DJNZ  R3,mDELAY1
            RET


;-------------------------------
;---------------------------------------
;------------------- END -------------------
;---------------------------------------
;-------------------------------
END
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
; FILENAME    N05.ASM ( ST.No 05 )
; DESCRIPTION  PROGRAM FOR Small Scale DCS (DIGITAL INPUT)
; HARDWARE    V-31
; ASSEMBLER   SXA51
; START-DATE  28/08/96
; SOFTWARE ENG  KMITL Power User
; COMPANY     KMITL , KMITNB


;********** Dim Variable ************

V0    EQU   30H
V1    EQU   31H
V2    EQU   32H
V3    EQU   33H
V4    EQU   34H
V5    EQU   35H
V6    EQU   36H
V7    EQU   37H

V_BlockST    EQU   40H
V_Block00    EQU   41H
V_Block01    EQU   42H
V_Block02    EQU   43H
V_Block03    EQU   44H
V_Block04    EQU   45H
V_Block05    EQU   46H
V_Block06    EQU   47H
V_Block07    EQU   48H
V_BlockEND   EQU   49H

;*****************************;
;    SERIAL USE INT
;       V-31
;*****************************;
    ORG   0000H
    AJMP  ST

    ORG   0023H  ;REM31 INT Vector RI+TI : MCS-51(0023H)
    JBC   RI,RECV                    ;TI+RI Vector Action
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    LJMP   OUT    ;if TI reti OUT


RECV:
    LCALL  RxBlock ;Save MV to V_Block
    mov    a,V_BlockST
    cjne   a,#21H ,OUT    ;!
    mov    a,V_Block00
    cjne   a,#30H ,OUT    ;0
    mov    a,V_Block01
    cjne   a,#35H ,OUT    ;5



    lcall  send
    JNB    TI,$
    CLR    TI
    SETB   ES

OUT:  RETI

SEND:
    setb   p3.4
    LCALL  PVIB   ;IN PV,Save to V_Block
    LCALL  CHKSUM
    MOV    V_Block07,A
    LCALL  TxBlock ;Tx V_Block
    clr    p3.4
    RET


    ORG    0100H
;****************** Set Const ****************
ST:         LCALL  INIv
          clr    p3.4
;SetIntV:     mov    dptr,#8008h            ;Not use for EPROM
;          mov    a,#80h                ;
;          movx   @dptr,a               ;
;          inc    dptr                ;
;          mov    a,#23h               ;
;          movx   @dptr,a               ;

OVER:
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    MOV   SCON,#50H                ;Serial Mode 1 9600
    ;MOV   PCON,#10000000B             ;x2 = 19200
    MOV   TMOD,#20H                ;
    MOV   TH1,#0FDH                ;
    SETB  TR1                      ;
    SETB  EA                       ;
    SETB  ES                       ;

                                   ;
        SJMP  $ ;add1              ; STOP

INIv:
    MOV   V_BlockST,#00100001B  ;!
    MOV   V_BlockEND,#00100011B  ;#

    MOV   V_Block00,#30H       ;0   ST.No.
    MOV   V_Block01,#35H       ;5   ST.No.
    MOV   V_Block02,#57H       ;W   General Command & Status
    MOV   V_Block03,#2BH       ;+   DATA H
    MOV   V_Block04,#2BH       ;+   DATA L
    MOV   V_Block05,#2BH       ;+   MultiPurpose
    MOV   V_Block06,#2BH       ;+   MultiPurpose
    MOV   V_Block07,#24H       ;$   ChkSUM

    RET

;MAIN1:



;************** TxBlock ************          ;Tx V_Block

                                   ;
TxBlock:    MOV   R0,#40H              ;
        LcaLL  mDELAY ;RS-485
TxBLoop:    MOV   A,@R0                ;
        LCALL  Tx1Byte             ;
        INC   R0                   ;
        CJNE  R0,#4AH,TxBLoop      ;
        RET                        ;

                                   ;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                                ;
; ********** Tx1Byte SUB **********                ;
; SEND 1 BYTE                                      ;
; IN = A                                           ;
; REG = NO                                         ;
                                ;
Tx1Byte:    CLR   TI                              ;
            CLR   ES                              ;
            MOV   SBUF,A                          ;
            JNB   TI,$                            ;
            RET                                   ;

;************** RxBlock ************
RxBlock:    LCALL  Rx1Byte
            ;MOV   V0,A
            mov  V_BlockST,A    ;SAVE 1ST from PC TO BLOCK
            cjne   a,#21H,RxOUT  ;!
        LCALL  Rx1Byte          ;ST.No. Byte 1
            ;MOV   V0,A
            mov  V_Block00,A    ;SAVE from PC TO BLOCK
            cjne   a,#30H,RxOUT  ;0
        LCALL  Rx1Byte          ;ST.No. Byte 2
            ;MOV   V0,A
            mov  V_Block01,A    ;SAVE from PC TO BLOCK
            cjne   a,#35H,RxOUT  ;
        LCALL  Rx1Byte          ;GSC Byte
            ;MOV   V0,A
        LCALL  Rx1Byte
            ;MOV   V0,A
            mov  V_Block03,A    ;SAVE MV from PC TO BLOCK
        LCALL  Rx1Byte
            ;MOV   V0,A
            mov  V_Block04,A    ;SAVE MV from PC TO BLOCK
        LCALL  Rx1Byte      ;+
            ;MOV   V0,A
        LCALL  Rx1Byte      ;+
            ;MOV   V0,A
        LCALL  Rx1Byte      ;ChkSum
            ;MOV   V0,A
RxOUT:      RET
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
; ********** Rx1Byte SUB **********
Rx1Byte:    MOV   A,SBUF
            JNB   RI,$          ;WAIT FOR RECEIVE OK
            CLR   RI
            RET

;**********PV IN & to V_Block SUB*************

PVIB:

            mov   a,p1                      ;Get input
            MOV   V1,A   ;SAVE INPUT TO V1          ;Save to
                                    ;V_Block
            MOV   A,V1                      ;
            LCALL  HTOA                     ;
             mov  V_Block03,r2  ;SAVE PV TO V_BLOCK    ;
             mov  V_Block04,r3  ;SAVE PV TO V_BLOCK    ;

                  RET


;******** HTOA SUB ***********
;CONVERT HEX TO ASCII
;IN  = A
;OUT = R2,R3
;REG = A,R2,R3
HTOA:   PUSH  ACC
            SWAP  A
            LCALL HTOAS
            MOV   R2,A
            POP   ACC
            LCALL HTOAS
            MOV   R3,A
            RET
HTOAS:  ANL   A,#0FH
            CJNE  A,#0AH,$+3
            JNC   HTOAS1
            ORL   A,#30H
            RET
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
HTOAS1:  SUBB A,#9
         ORL  A,#40H
         RET


; ********** ATOH SUB **********
; ASCII TO HEX CONVERT
; IN  = R2,R3  30H,41H
; OUT = A      0AH
; REG = A,R2

ATOH:  MOV   A,R2
       LCALL ATOHS
       SWAP  A
       MOV   R2,A
       MOV   A,R3
       LCALL ATOHS
       ORL   A,R2
       RET
ATOHS: CJNE  A,#'A',$+3
       JC    ATOHS1
       ADD   A,#9
ATOHS1: ANL  A,#0FH
       RET


; ********** CHKSUM SUB **********
;Sum V_Block00-06 without carry
;IN = V_Block00-06
;OUT = A

CHKSUM:
         MOV   A,#00H
         ADD   A,V_Block00
         ADD   A,V_Block01
         ADD   A,V_Block02
         ADD   A,V_Block03
         ADD   A,V_Block04
         ADD   A,V_Block05
         ADD   A,V_Block06
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        RET
; ********** DELAY SUB **********
; DELAY SUBROUTINE
; IN = R2
; REG = R2,R3,R4

DELAY:      MOV   R3,#0
DELAY1:     MOV   R4,#0
                  DJNZ  R4,$
                  DJNZ  R3,DELAY1
                  DJNZ  R2,DELAY
                  RET

; ********** mDELAY SUB **********
; mDELAY SUBROUTINE
; REG = R3,R4

mDELAY:     MOV   R3,#3FH
mDELAY1:    MOV   R4,#0
                  DJNZ  R4,$
            DJNZ  R3,mDELAY1
                  RET


;-------------------------------
;---------------------------------------
;---------------------- END ----------------------
;---------------------------------------
;-------------------------------
END
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
; FILENAME    N06.ASM
; DESCRIPTION  PROGRAM FOR Small Scale DCS (Digital OUTPUT)
; HARDWARE    V-31
; ASSEMBLER   SXA51
; START-DATE  28/08/96
; SOFTWARE ENG  KMITL Power User
; COMPANY     KMITL , KMITNB


;********** Dim Variable **********

V0    EQU   30H
V1    EQU   31H
V2    EQU   32H
V3    EQU   33H
V4    EQU   34H
V5    EQU   35H
V6    EQU   36H
V7    EQU   37H

V_BlockST    EQU   40H
V_Block00    EQU   41H
V_Block01    EQU   42H
V_Block02    EQU   43H
V_Block03    EQU   44H
V_Block04    EQU   45H
V_Block05    EQU   46H
V_Block06    EQU   47H
V_Block07    EQU   48H
V_BlockEND   EQU   49H

;*****************************;
;    SERIAL USE INT
;     ANT-32,REM31
;*****************************;
          ORG   0000H
    CLR   EA    ;All Interrupt Disable
          AJMP  ST

          ORG   0023H  ;REM31 INT Vector RI+TI : MCS-51(0023H)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        JBC   RI,RECV                ;TI+RI Vector Action
        LJMP  OUT   ;if TI reti OUT

RECV:

    PUSH  00H
    PUSH  02H
    PUSH  03H
    PUSH  04H
    PUSH  05H
    PUSH  06H
    PUSH  07H
    PUSH  ACC
    PUSH  20H


        LCALL  RxBlock  ;Save MV to V_Block
        mov    a,V_BlockST
        cjne   a,#21H ,OUT   ;!
        mov    a,V_Block00
        cjne   a,#30H ,OUT   ;0
        mov    a,V_Block01
        cjne   a,#36H ,OUT   ;6

    MOV   A,V_Block02   ;GSC
    CJNE  A,#57H,NoMVO   ;W
    LCALL  MVO    ;Get MV from Block,Convert to HEX,OUT PORT1
NoMVO:
        lcall  send
        JNB   TI,$
        CLR   TI
        SETB  ES


OUT:
    POP   20H
    POP   ACC
    POP   07H
    POP   06H
    POP   05H
    POP   04H
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    POP   03H
    POP   02H
    POP   00H


    RETI

SEND:
          setb  p3.4
          LCALL CHKSUM
          MOV   V_Block07,A
          LCALL TxBlock ;Tx V_Block
          clr   p3.4
          RET

          ORG   0100H
;***************** Set Const ****************
ST:       LCALL INIv
          MOV   SP,#50H
                clr   p3.4
;SetIntV:    mov   dptr,#8008h          ;Not use for EPROM
;         mov   a,#80h              ;
;         movx  @dptr,a             ;
;         inc   dptr                ;
;         mov   a,#23h              ;
;         movx  @dptr,a             ;

OVER:
          MOV   SCON,#50H                ;Serial Mode 1 9600
          ;MOV   PCON,#10000000B          ;x2 = 19200
          MOV   TMOD,#20H                ;
          MOV   TH1,#0FDH                ;
          SETB  TR1                      ;
          SETB  EA                       ;
          SETB  ES                       ;
    MOV   20H,#10001000B

ROT:  MOV   A,V_Block02   ;GSC
    CJNE  A,#57H,ROTR   ;W
    LJMP  STOP
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
ROTR:  CJNE   A,#52H,ROTL   ;R
       MOV    A,20H
       MOV    P1,A
       RR     A
       MOV    20H,A
       LCALL  DYs
       LJMP   ROT
ROTL:  CJNE   A,#4CH,STOP   ;L
       MOV    A,20H
       MOV    P1,A
       RL     A
       MOV    20H,A
       LCALL  DYs
       LJMP   ROT
STOP:
       LJMP   ROT

;********* DELAY_SL SUB **********;;DELAY FOR INC AND DEC SLOW
DYs:
       MOV    R2,V_Block03
       MOV    R3,V_Block04
       LCALL  ATOH
       MOV    R5,A
DY9:   MOV    R7,A
DY8:   MOV    R6,A
       DJNZ   R6,$
       DJNZ   R7,DY8
       DJNZ   R5,DY9
       RET

INIv:
            MOV    V_BlockST,#00100001B   ;!
            MOV    V_BlockEND,#00100011B  ;#

            MOV    V_Block00,#30H      ;0   ST.No.
            MOV    V_Block01,#36H      ;6   ST.No.
            MOV    V_Block02,#57H      ;W   General Command & Status
            MOV    V_Block03,#2BH      ;+   DATA H
            MOV    V_Block04,#2BH      ;+   DATA L
            MOV    V_Block05,#2BH      ;+   MultiPurpose
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        MOV   V_Block06,#2BH      ;+   MultiPurpose
        MOV   V_Block07,#24H      ;$   ChkSUM

        RET

;MAIN1:



;************** TxBlock *************          ;Tx V_Block
                                                ;
TxBlock:    MOV   R0,#40H             ;
        Lcall  mDELAY ; RS-485
TxBLoop:    MOV   A,@R0               ;
                LCALL  Tx1Byte            ;
                INC   R0                  ;
                CJNE  R0,#4AH,TxBLoop     ;
                RET                       ;
                                          ;
                                          ;
; ********** Tx1Byte SUB **********       ;
; SEND 1 BYTE                   ;
; IN  = A                       ;
; REG = NO                      ;
                                          ;
Tx1Byte:    CLR   TI                  ;
                CLR   ES                  ;
                MOV   SBUF,A              ;
                JNB   TI,$                ;
                RET                       ;

;************** RxBlock ************
RxBlock:    LCALL  Rx1Byte
                ;MOV   V0,A
                mov  V_BlockST,A    ;SAVE 1ST from PC TO BLOCK
                cjne  a,#21H,RxOUT   ;!
        LCALL  Rx1Byte            ;ST.No. Byte 1
                ;MOV   V0,A
                 mov  V_Block00,A    ;SAVE from PC TO BLOCK
                 cjne  a,#30H,RxOUT  ;0
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        LCALL  Rx1Byte          ;ST.No. Byte 2
               ;MOV    V0,A
               mov  V_Block01,A    ;SAVE from PC TO BLOCK
               cjne  a,#36H,RxOUT  ;6
        LCALL  Rx1Byte          ;GSC Byte
               ;MOV    V0,A
               mov  V_Block02,A    ;SAVE MV from PC TO BLOCK
        LCALL  Rx1Byte
               ;MOV    V0,A
               mov  V_Block03,A    ;SAVE MV from PC TO BLOCK
        LCALL  Rx1Byte
               ;MOV    V0,A
               mov  V_Block04,A    ;SAVE MV from PC TO BLOCK
        LCALL  Rx1Byte      ;+
               ;MOV    V0,A
        LCALL  Rx1Byte      ;+
               ;MOV    V0,A
        LCALL  Rx1Byte      ;ChkSum
               ;MOV    V0,A
RxOUT:     RET
; ********** Rx1Byte SUB **********
Rx1Byte:
        MOV   A,SBUF
               JNB   RI,$          ;WAIT FOR RECEIVE OK
               CLR   RI
               RET


;**********MV OUT SUB**********

MVO:

               mov   r2,v_block03          ;MV from V_Block
               mov   r3,v_block04          ;
               lcall  atoh  ;R2,R3 -> A          ;Convert to HEX
               mov   v2,a                  ;Save to V2

               MOV    A,V2                       ;OUT MV to Port1
               MOV   p1,A                  ;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                RET

;******** HTOA SUB ***********
;CONVERT HEX TO ASCII
;IN  = A
;OUT = R2,R3
;REG = A,R2,R3
HTOA:   PUSH  ACC
        SWAP  A
        LCALL HTOAS
        MOV   R2,A
        POP   ACC
        LCALL HTOAS
        MOV   R3,A
        RET
HTOAS:  ANL   A,#0FH
        CJNE  A,#0AH,$+3
        JNC   HTOAS1
        ORL   A,#30H
        RET
HTOAS1: SUBB  A,#9
        ORL   A,#40H
        RET


; ********** ATOH SUB **********
; ASCII TO HEX CONVERT
; IN  = R2,R3   30H,41H
; OUT = A       0AH
; REG = A,R2

ATOH:   MOV   A,R2
        LCALL ATOHS
        SWAP  A
        MOV   R2,A
        MOV   A,R3
        LCALL ATOHS
        ORL   A,R2
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        RET
ATOHS:  CJNE  A,#'A',$+3
        JC    ATOHS1
        ADD   A,#9
ATOHS1: ANL   A,#0FH
        RET


; ********** CHKSUM SUB **********
;Sum V_Block00-06 without carry
;IN = V_Block00-06
;OUT = A

CHKSUM:
                MOV   A,#00H
                ADD   A,V_Block00
                ADD   A,V_Block01
                ADD   A,V_Block02
                ADD   A,V_Block03
                ADD   A,V_Block04
                ADD   A,V_Block05
                ADD   A,V_Block06

                RET
; ********** DELAY SUB **********
; DELAY SUBROUTINE
; IN  = R2
; REG = R2,R3,R4

;DELAY:      MOV   R3,#0
;DELAY1:     MOV   R4,#0
;           DJNZ  R4,$
;           DJNZ  R3,DELAY1
;           DJNZ  R2,DELAY
;           RET
; ********** mDELAY SUB **********
; mDELAY SUBROUTINE
; REG = R3,R4

mDELAY:     MOV   R3,#3FH
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
mDELAY1:    MOV   R4,#0
            DJNZ  R4,$
        DJNZ  R3,mDELAY1
            RET

;-------------------------------
;-----------------------------------------
;------------------------ END ------------------------
;-----------------------------------------
;-------------------------------
END
```

038016

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้