



ปีการศึกษา 2532

ตัวควบคุมออกฤทธิ์แบบช่วงชนิดโปรแกรมได้



อจจารย์ วันชัย รุ่งรุจา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวควบคุมอุณหภูมิแบบช่วงชนิดโปรแกรมได้
PROGRAMMABLE RANGED TEMPERATURE CONTROLLER



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

026908

ปริญญาโทปีการศึกษา 2532

ภาควิชา วิศวกรรมระบบควบคุม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง หัวควบคุมอุณหภูมิแบบช่วงชนิดโปรแกรมได้

ผู้จัดทำ

1. นาย ชัยพร รัตนเชตกุล
2. นาย ศรัณย์ ศรีพิพัฒน์



(อาจารย์ วันชัย รีรุจา)

อาจารย์ที่ปรึกษา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวควบคุมอุณหภูมิแบบช่วงชนิดโปรแกรมได้

ชัยพร รัตน์เชตกุล

ศรัณย์ ศรีนิพนธ์

วันชัย รุ่งรุจา อาจารย์ที่ปรึกษา

ปีการศึกษา 2532

บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้ เป็นการนำไมโครคอนโทรลเลอร์ตระกูล เอ็มซีเอส 51 (MCS-51) เบอร์ 8031 มาใช้เป็นตัวควบคุมอุณหภูมิของขบวนการย้อมผ้า โดยแบ่งการควบคุมอุณหภูมิในขบวนการออกเป็นช่วง ๆ การทำงานของตัวควบคุม จะถูกกำหนดโดยโปรแกรม ซึ่งอยู่ภายในหน่วยความจำ มีการรับข้อมูลการควบคุมจากผู้ใช้ และมีการแสดงค่าการควบคุมในส่วนแสดงผล

PROGRAMMABLE RANGED TEMPERATURE CONTROLLER

CHAIYAPORN RATANACHETKUL

SRAN SRIBHIBHADH

WANCHAI RIEWRUJA ADVISOR

1989

ABSTRACT

This Thesis, programmable ranged temperature controller, is the application of Microcontroller MCS-51 number 8031 utilized to control temperature. In temperature controlling, temperature of process will be apportioned to the sub-time range. By program in memory unit receiving data from user and showing the result on display, the operation of the controller will be set.

สารบัญ

บทนำ	1
บทที่ 1 ทฤษฎี	2
บทที่ 2 ไมโครคอนโทรลเลอร์ 8031	12
บทที่ 3 หลักการทำงาน	30
บทที่ 4 การทดลอง และ ผลการทดลอง	45
บทที่ 5 สรุป และ วิจัย	53
ภาคผนวก ก	55
ภาคผนวก ข	61
กิตติกรรมประกาศ	65
หนังสืออ้างอิง	66



บทนำ

ในปัจจุบัน มีการนำเอาไมโครคอมพิวเตอร์มาใช้งานในด้านต่าง ๆ มากมาย ได้แก่ การควบคุมการทำงานของเครื่องใช้ไฟฟ้าภายในบ้าน ไปจนถึงการควบคุมขบวนการที่ซับซ้อนในโรงงานอุตสาหกรรม โดยในการทำงานใช้ไมโครโปรเซสเซอร์เป็นหน่วยประมวลผลกลาง และทำงานประกอบกับส่วนอื่น ๆ อันได้แก่ หน่วยความจำ รอม (ROM) และ แรม (RAM) หน่วยพอร์ตอินพุท และ เอาท์พุท

แต่ในลักษณะของการควบคุมเฉพาะอย่าง เช่น การควบคุมการทำงานของเครื่องมือวัดชนิดต่าง ๆ การนำเครื่องไมโครคอมพิวเตอร์ทั้งเครื่องมาใช้ควบคุมการทำงานอาจเกิดความไม่สะดวกในการเคลื่อนย้าย และ อาจไม่คุ้มกับการลงทุน จึงได้เกิดการนำเอาหน่วยต่าง ๆ ของไมโครคอมพิวเตอร์เฉพาะที่จำเป็นสำหรับการควบคุมชนิดนั้นๆ มารวมกันเป็นหน่วยเดียว ทำให้มีขนาดลดลงเหลือขนาดเท่าชิปไมโครโปรเซสเซอร์ทั่วไป โดยถูกนำไปใช้ในงานด้านต่าง ๆ เป็นหลัก จึงมีชื่อเรียกใหม่ว่า ไมโครคอนโทรลเลอร์ (Microcontroller)

ไมโครคอนโทรลเลอร์ตระกูลที่ได้รับความนิยมใช้กันอย่างแพร่หลายตระกูลหนึ่งคือ ตระกูลเอ็มซีเอส 51 (MCS-51) เพราะเป็นตระกูลที่มีขีดความสามารถในการทำงานสูง มีราคาถูก และสามารถซื้อหาได้ง่ายตามท้องตลาด ในการทำโครงการนี้จึงเลือกไมโครคอนโทรลเลอร์ตระกูลนี้ (เบอร์ 8031) มาใช้ในการออกแบบสร้างตัวควบคุมสำหรับการควบคุมอุณหภูมิของขบวนการย้อมสีผ้าในโรงงานอุตสาหกรรม

1.1 การควบคุมขบวนการ (process control)

ในปัจจุบัน อุตสาหกรรมการผลิตได้มีการขยายตัวขึ้นอย่างรวดเร็ว นับวันจะมีผลิตภัณฑ์ใหม่ ๆ บ่อยเข้าสู่ตลาดการบริโภค คุณภาพของผลิตภัณฑ์ จึงเป็นสิ่งสำคัญเพราะถ้าผลิตภัณฑ์ที่มีคุณภาพไม่ดี ไม่เป็นที่พอใจของผู้ที่บริโภค ก็ย่อมไม่สามารถแข่งขันในตลาดได้ ดังนั้น การควบคุมขบวนการผลิตที่แน่นอนและแม่นยำจึงเป็นสิ่งจำเป็น

คุณสมบัติของผลิตภัณฑ์จะขึ้นกับค่าตัวแปรต่าง ๆ ในขบวนการ ตัวอย่างเช่น อุณหภูมิ ความดัน อัตราการไหล จุดมุ่งหมายของการควบคุมขบวนการก็คือ การควบคุมตัวแปรต่าง ๆ ในขบวนการเหล่านี้ได้อย่างเหมาะสม เพื่อให้ได้คุณสมบัติของผลิตภัณฑ์เป็นไปตามที่ต้องการ ซึ่งตัวแปรที่จะต้องควบคุมนี้ อาจมีเพียงตัวแปรเดียว หรือหลายตัวแปรขึ้นอยู่กับ ความซับซ้อนของขบวนการที่เราทำการควบคุม

การควบคุมขบวนการ จะประกอบด้วยขั้นตอนต่าง ๆ ดังนี้ คือ

1. เลือกตัวแปรของขบวนการที่จะนำมาควบคุม
2. ทำการวัดค่าตัวแปรที่ควบคุมนั้น
3. เปรียบเทียบค่าที่วัดได้กับค่าอ้างอิง (set point)

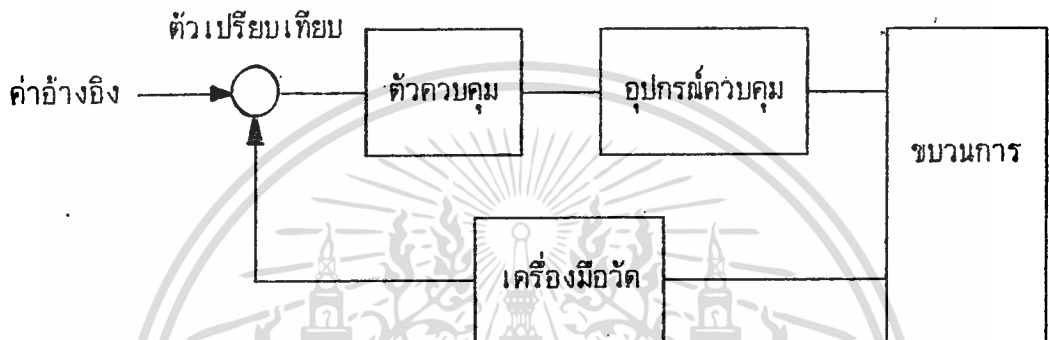
4. กำหนดการเปลี่ยนแปลงของตัวแปรควบคุม เพื่อที่จะแก้ไข การเบี่ยงเบนของตัวแปรควบคุมจากค่าที่ตั้งไว้

5. บ้อนกลับค่าการเปลี่ยนแปลงของ ตัวแปรควบคุมกลับไปยังขบวนการ เพื่อที่จะแก้ไขค่าตัวแปรควบคุมในขบวนการ ให้ถูกต้อง

โดยขั้นตอนที่สองถึงห้า จะถูกวนกระทำซ้ำไปเรื่อย ๆ ตลอดการควบคุม โดยมีการอ้างอิงกับเวลา

1.2 รูปของการควบคุมขบวนการ (process control loop)

เพื่อให้เข้าใจถึงการทำงานของระบบควบคุมขบวนการ เราสามารถอธิบายได้ด้วย บล็อกไดอะแกรมดังรูปที่ 1.1



รูปที่ 1.1

บล็อกไดอะแกรมแสดงรูปการควบคุมขบวนการ

แต่ละบล็อกสามารถอธิบายได้ดังนี้

ขบวนการ คือ ขบวนการทั้งหมดที่จะควบคุม ประกอบด้วย ตัวแปรมากมายที่มีอิทธิพลต่อขบวนการ เมื่อตัวแปรเกิดการเปลี่ยนแปลง ก็จะมีผลกระทบต่อคุณสมบัติของผลิตภัณฑ์

เครื่องมือวัด คือ ส่วนที่ทำกรวัดค่าตัวแปรควบคุมจากขบวนการ โดยส่วนเอาต์พุต (output) ของเครื่องมือวัดก็คือ ค่าการวัดที่มีรูปแบบต่าง ๆ เช่น ค่าความต่างศักย์ กระแสไฟฟ้า หรือสัญญาณดิจิทัล

ตัวเปรียบเทียบ คือ ส่วนที่กำหนดการเปลี่ยนแปลงของตัวแปรควบคุม เพื่อที่จะจัดค่าความเบี่ยงเบนของตัวแปรควบคุม เอาท์พุทของตัวควบคุม จะถูกป้อนกลับสู่ขบวนการ เพื่อแปรค่าตัวแปรควบคุมให้เป็นไปตามที่กำหนด

อุปกรณ์ควบคุม (control element) คือ ส่วนที่ทำการเปลี่ยนค่าสัญญาณจากตัวควบคุมไปสู่ค่าตัวแปรของขบวนการ

การทำงานของลูประจะเริ่มจาก ค่าตัวแปรควบคุมในขบวนการ จะถูกวัดจากเครื่องมือวัด และจะถูกเปรียบเทียบกับค่าอ้างอิง จากนั้น ค่าผิดพลาดจะถูกส่งเข้าไปยังตัวควบคุม โดยตัวควบคุมก็จะคำนวณเพื่อแก้ค่าผิดพลาด แล้วส่งสัญญาณไปยังอุปกรณ์ควบคุม เพื่อเปลี่ยนแปลงค่าตัวแปรควบคุมในขบวนการให้เป็นไปตามค่าที่ต้องการ และจะกระทำซ้ำเช่นนี้เรื่อยไป

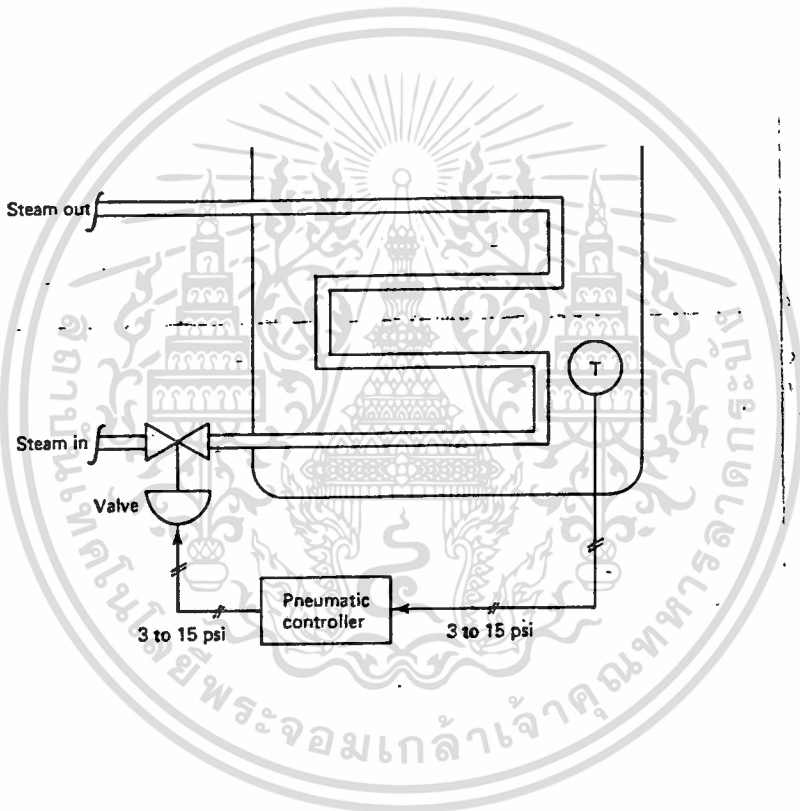
1.3 รูปแบบของการควบคุมขบวนการ

1.3.1 การควบคุมขบวนการแบบอนาลอก

ค่าตัวแปรควบคุมในขบวนการ จะถูกเปลี่ยนเป็นสัญญาณ ซึ่งสัญญาณนี้จะใช้ใน ตัวควบคุมในการสร้างสัญญาณเอาท์พุท เพื่อไปขับอุปกรณ์ควบคุม ซึ่งจะ เป็นผลให้ตัวแปรควบคุมในขบวนการเปลี่ยนแปลง สัญญาณที่ใช้ควบคุมโดยทั่วไปจะอยู่ในรูปของความดันอากาศ หรือกระแสไฟฟ้า โดยเรียกว่า ระบบการควบคุมแบบนิวแมติก (pneumatic process control system) และ ระบบการควบคุมแบบอิเล็กทรอนิกส์ (electronic process control system) ตามลำดับ

ในระบบควบคุมแบบนิวแมติก ค่าตัวแปรของขบวนการ จะถูกแทนด้วยสัญญาณความดัน 3-15 ปอนด์ต่อตารางนิ้ว (psi) โดยค่าต่ำสุดของตัวแปรของขบวนการจะถูกแทนด้วยความดัน 3 ปอนด์ต่อตารางนิ้ว และค่าสูงสุดแทนด้วย 15 ปอนด์ต่อตารางนิ้ว ส่วนสัญญาณเอาท์พุทจากตัวควบคุม 0 เปอร์เซ็นต์ ก็จะถูกแทนด้วย 3 ปอนด์ต่อตารางนิ้ว และ 100 เปอร์เซ็นต์ ถูกแทนด้วย 15 ปอนด์ต่อตารางนิ้ว

ตัวอย่างของระบบ การควบคุมแบบนิวแมติกนี้ แสดงได้ดังรูป 1.2 ซึ่งเป็นระบบควบคุมอุณหภูมิที่มีอุณหภูมิเป็นตัวแปรควบคุม อินพุทของขบวนการคือ ไอน้ำที่ไหลเข้าขบวนการโดยผ่านวาล์ว เครื่องมือวัด คือ เทอร์โมมิเตอร์แบบใช้ความดันก๊าซ (gas pressure thermometer) ซึ่งจะเปลี่ยนการเปลี่ยนแปลงของอุณหภูมิไปเป็นค่าความดันต่าง ๆ โดยตัวควบคุมเป็นอนาล็อกคอมพิวเตอร์แบบนิวแมติก และอุปกรณ์ควบคุมคือ วาล์วควบคุม ซึ่งควบคุมการปิด-เปิด โดยค่าความดันจากตัวควบคุม



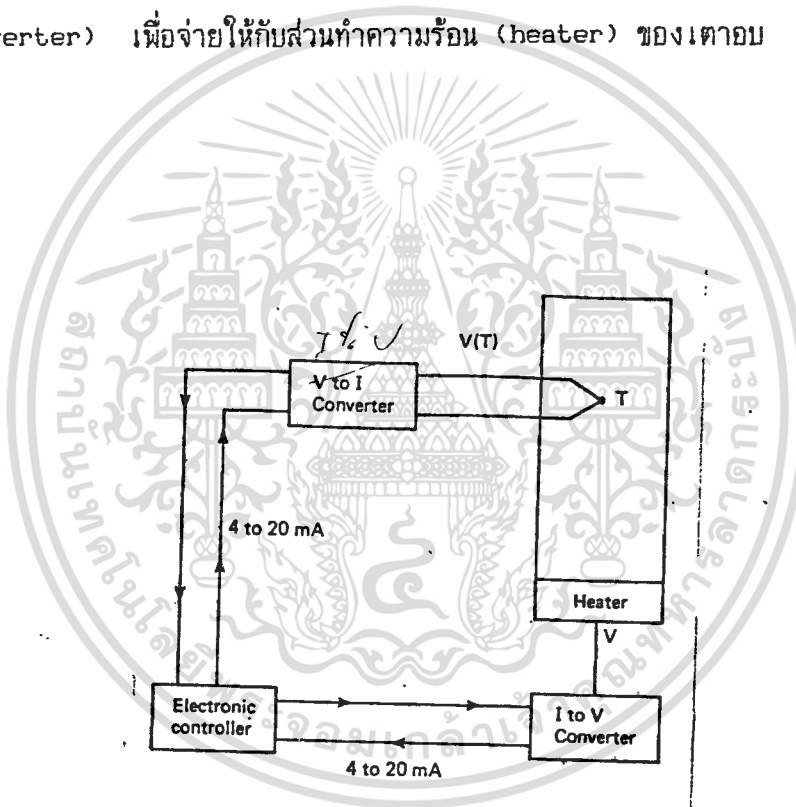
รูปที่ 1.2

ระบบควบคุมอุณหภูมิ

ส่วนระบบควบคุมแบบอิเล็กทรอนิกส์ ที่ใช้กระแสไฟฟ้าเป็นสัญญาณควบคุม ตัวแปรของขบวนการและสัญญาณ เอาท์พุทจากตัวควบคุม จะถูกแทนด้วยค่ากระแสตรง 4-20

มิลลิแอมป์ โดยที่ค่าต่ำสุดจะถูกแทนด้วยค่ากระแส 4 มิลลิแอมป์ และค่าสูงสุดถูกแทนด้วยค่ากระแส 20 มิลลิแอมป์

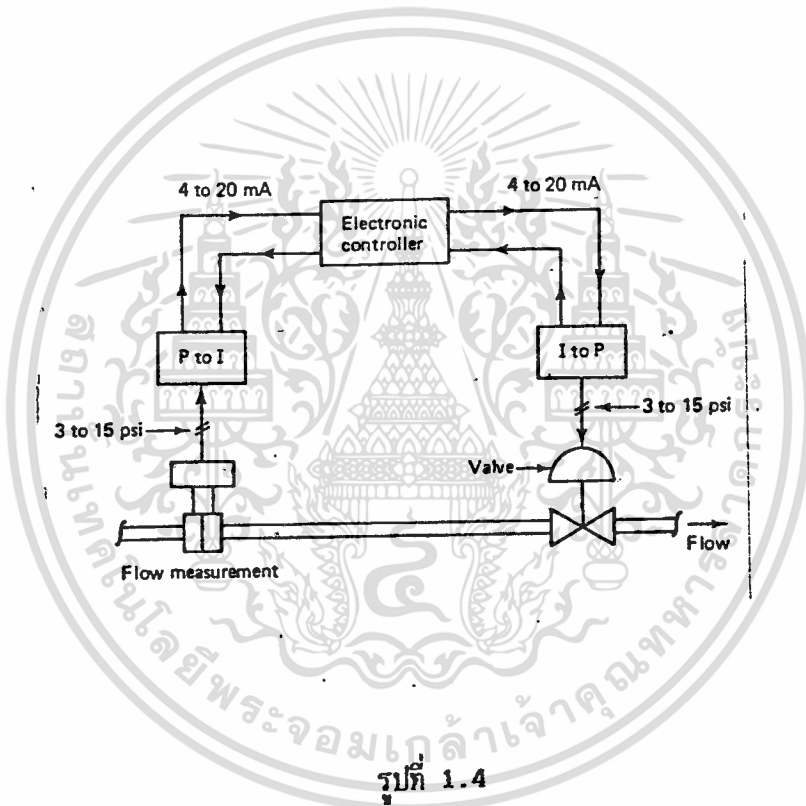
ตัวอย่างของระบบการควบคุมแบบอิเล็กทรอนิกส์ แสดงได้ดังรูปที่ 1.3 ซึ่งเป็นระบบควบคุมอุณหภูมิในเตาอบโดยใช้คู่ความความร้อน (thermocouple) เป็นตัววัดค่าอุณหภูมิ แล้วส่งค่าที่วัดผ่านวงจรแปลงศักดาเป็นกระแส (V to I converter) เป็นค่ากระแส 4-20 มิลลิแอมป์ และค่ากระแสนี้ จะถูกส่งไปยังตัวควบคุม จากนั้น ตัวควบคุมจะส่งสัญญาณกระแส 4-20 มิลลิแอมป์ ผ่านวงจรแปลงกระแสเป็นค่าศักดาไฟฟ้า (I to V converter) เพื่อจ่ายให้กับส่วนทำความร้อน (heater) ของเตาอบ



รูปที่ 1.3
ระบบควบคุมอุณหภูมิในเตาอบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นอกจากนี้ ยังมีระบบการควบคุมที่ใช้สัญญาณทั้งแบบนิวแมติก และอิเล็กทรอนิกส์ ร่วมกัน ตัวอย่างดังรูป 1.4 ซึ่งเป็นการควบคุมอัตราการไหลในท่อ โดยใช้ความดันในการเปิด-ปิดวาล์วควบคุม แต่ใช้ตัวควบคุมเป็นอิเล็กทรอนิกส์ ซึ่งมีตัวแปลงความดันให้เป็นกระแส (P to V converter) และ ตัวแปลงกระแสให้เป็นความดัน (I to P converter) ที่อินพุทและเอาท์พุทของตัวแปรควบคุม

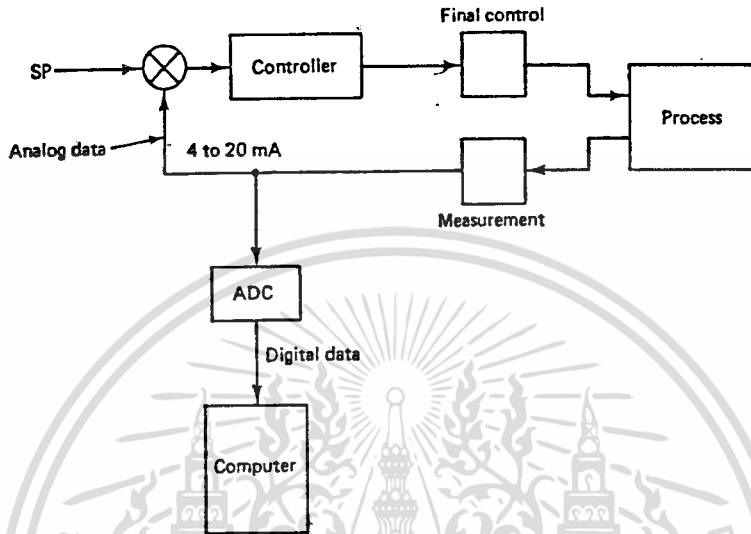


รูปที่ 1.4
การควบคุมอัตราการไหลในท่อ

1.3.2 การควบคุมขบวนการแบบที่ใช้คอมพิวเตอร์สนับสนุน

ในสมัยก่อนที่ยังไม่มีคอมพิวเตอร์ การควบคุมขบวนการจะใช้ในแบบอนาลอกทั้งหมด ต่อมาเมื่อมีการพัฒนาทางด้านคอมพิวเตอร์เกิดขึ้น จึงเกิดการนำคอมพิวเตอร์มา

ประยุกต์ใช้กับการควบคุมขบวนการ ตัวอย่างเช่น การนำคอมพิวเตอร์มาใช้ในการเก็บข้อมูล และนำข้อมูลมาวิเคราะห์ผลการควบคุมที่ผ่านมา ดังรูปที่ 1.5



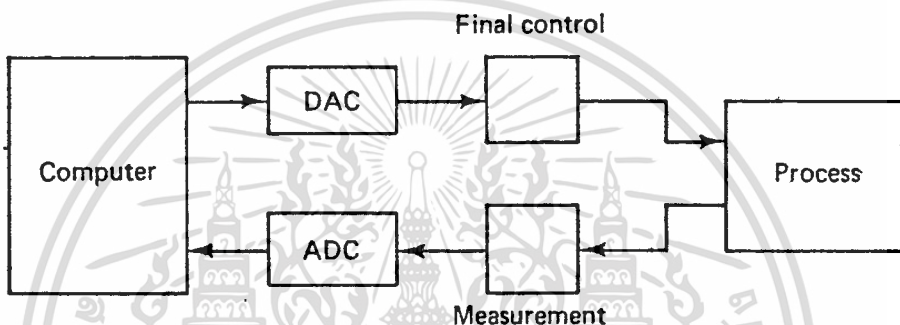
รูปที่ 1.5

การใช้คอมพิวเตอร์ในการเก็บและวิเคราะห์ข้อมูล

1.3.3 การควบคุมขบวนการแบบดิจิทัลโดยตรง (direct digital control)

เมื่อคอมพิวเตอร์ได้พัฒนาขึ้นมาจนการคำนวณมีความแม่นยำและความรวดเร็วสูง จึงถูกนำมาใช้ในการควบคุมขบวนการ โดยแทนที่ตัวควบคุมแบบอนาลอก โดยใช้วงจรแปลงสัญญาณดิจิทัลเป็นอนาลอก (digital to analog converter) และวงจรแปลงสัญญาณอนาลอกเป็นดิจิทัล (analog to digital converter) เป็นสื่อกลางการติดต่อระหว่างตัวควบคุมและขบวนการ โดยค่าสัญญาณที่วัดได้จากขบวนการ จะถูกแปลงเป็นสัญญาณดิจิทัล เพื่อประมวลผลในคอมพิวเตอร์ และผลจากการประมวล ก็จะ

ถูกแปลงกลับมาเป็นสัญญาณอนาลอก เพื่อใช้ในการควบคุมขบวนการต่อไป การใช้การควบคุมขบวนการแบบดิจิทัลโดยตรงนี้ จะสามารถกำหนดการทำงานของตัวควบคุมได้จากซอฟต์แวร์ของคอมพิวเตอร์ ซึ่งสะดวกในการเปลี่ยนแปลงแก้ไขการควบคุม



รูปที่ 1.6

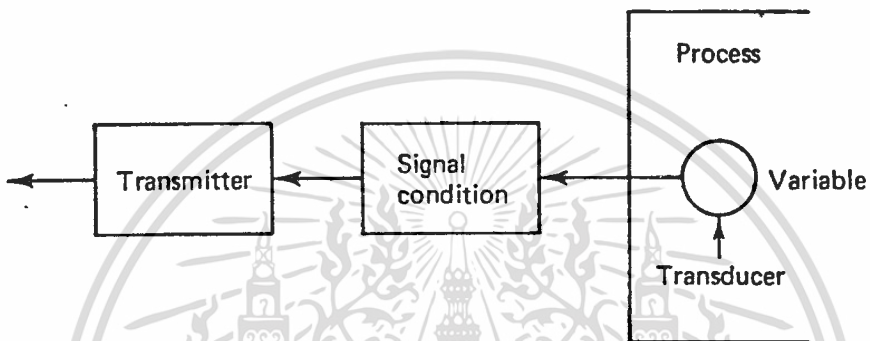
การควบคุมขบวนการแบบดิจิทัลโดยตรง

1.4 การวัดค่าตัวแปรจากขบวนการ

ในรูปการควบคุมขบวนการ สิ่งสำคัญสิ่งหนึ่งที่จะต้องพิจารณาคือ การวัดค่าตัวแปรที่เราจะควบคุม ในระบบการควบคุมแบบอนาลอก ค่าที่วัดได้จะต้องถูกปรับแต่งอย่างเหมาะสม ก่อนที่จะนำไปสู่ส่วนของตัวควบคุม ส่วนในระบบการควบคุมแบบอนาลอก

ก็จะต้องมีการปรับแต่งสัญญาณ ให้เหมาะสมเช่นเดียวกัน ก่อนที่จะทำการแปลงเป็นสัญญาณดิจิทัล เพื่อประมวลผล

การวัดค่าตัวแปรขบวนการ จะประกอบด้วยส่วนต่าง ๆ ดังรูปที่ 1.7



รูปที่ 1.7

การวัดค่าตัวแปรขบวนการ

จากรูปสามารถอธิบายส่วนต่าง ๆ ได้ดังนี้

ทรานสดิวเซอร์ (transducer) เป็นส่วนที่แปลงค่าตัวแปรควบคุมจากขบวนการ เป็นค่าสัญญาณที่เหมาะสมที่จะได้ใช้ต่อไป ซึ่งมักจะอยู่ในรูปของสัญญาณไฟฟ้า ฟังก์ชันถ่ายโอน (transfer function) ของทรานสดิวเซอร์ จะเป็นตัวกำหนดความสัมพันธ์ระหว่างตัวแปรควบคุม และค่าสัญญาณเอาต์พุทของทรานสดิวเซอร์

ส่วนปรับแต่งสัญญาณ (signal conditioning) ค่าสัญญาณเอาต์พุทจากทรานสดิวเซอร์ จะถูกแปลงเป็นค่าสำหรับนำไปใช้ในส่วนต่อไป ตัวอย่างเช่น ในระบบ

ความคุมขบวนการแบบคอมพิวเตอรื ค่าสัญญาณที่วัดโดยทรานสดิวเซอร์ จะถูกเปลี่ยนเป็นค่าที่เหมาะสมก่อนที่จะนำไปเข้าส่วนที่แปลง เป็นสัญญาณดิจิทัล

ทรานสมิตเตอร์ (transmitter) ในกรณีที่ขบวนการและส่วนควบคุมไม่ได้ อยู่ที่เดียวกัน ก็จำเป็นที่จะต้องมีส่วนทรานสมิตเตอร์นี้ขึ้นมา เพื่อแปลงสัญญาณที่วัดได้เป็นสัญญาณที่จะส่งไปยังตัวควบคุม โดยปกติมักนิยมการใช้การส่งในรูปกระแสไฟฟ้า หรืออาจ จะแปลงเป็นสัญญาณดิจิทัลก่อน แล้วส่งสัญญาณออกไปแบบอนุกรม



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ไมโครคอนโทรลเลอร์ 8031

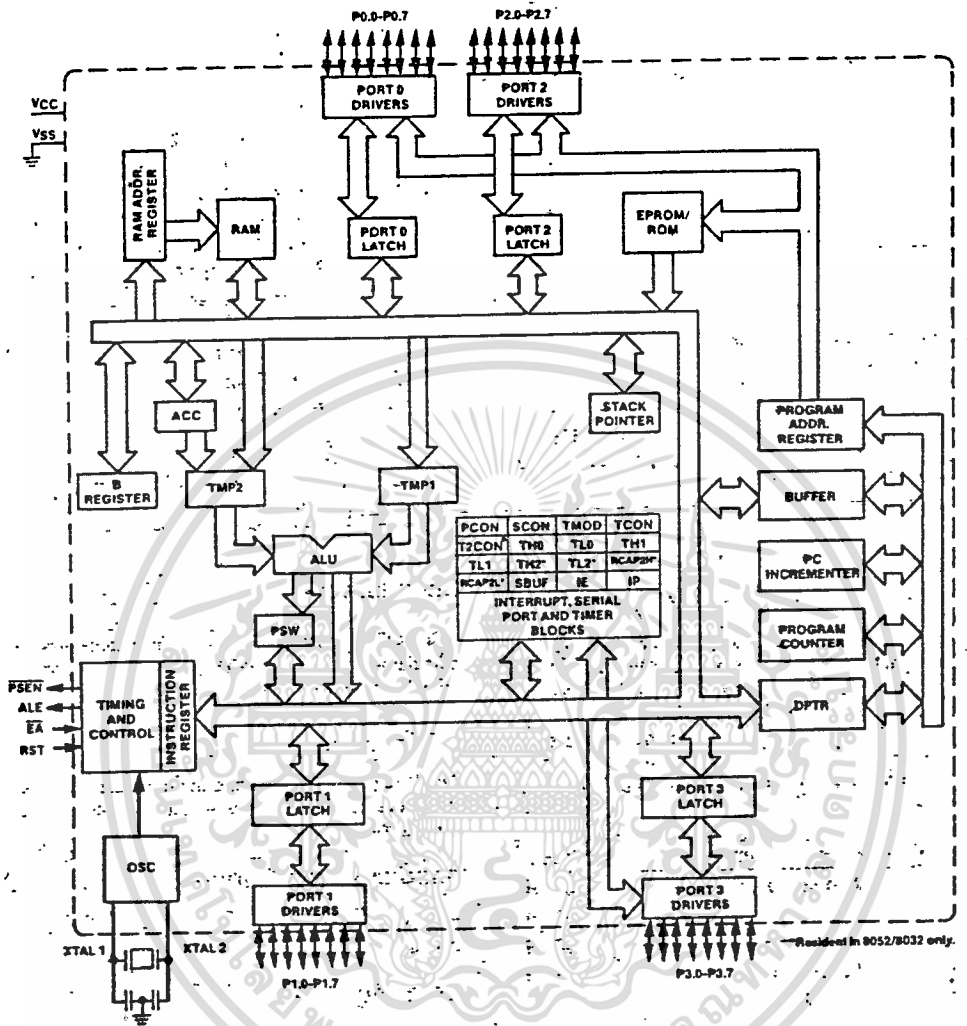
เป็นส่วนควบคุมการทำงานทั้งหมดในโครงงานนี้ มีลักษณะเหมือนไมโครโปรเซสเซอร์ทั่วไป และเพิ่มอุปกรณ์ต่าง ๆ ที่สำคัญลงไป ทำให้ 8031 มีความสามารถมาก

ลักษณะที่สำคัญของ 8031 คือ

- ไมโครโปรเซสเซอร์แบบ 8 บิต
- มีวงจรสร้างสัญญาณนาฬิกาภายใน
- 32 อินพุต / 32 เอาต์พุต
- อ้างอิงหน่วยความจำภายนอกสำหรับเก็บข้อมูล 64 กิโลไบต์
- อ้างอิงหน่วยความจำภายนอกสำหรับเก็บโปรแกรม 64 กิโลไบต์
- มีวงจรมับ, ตั้งเวลา 16 บิต 2 ชุด
- 5 อินเทอร์รัพท์ สามารถจัดลำดับความสำคัญได้ 2 ระดับ
- ส่งข้อมูลอนุกรมแบบ 2 ทิศทางได้
- สามารถประมวลตรรกศาสตร์ได้
- มี RAM ภายในขนาด 128 ไบต์ โดยสามารถอ้างอิงได้ทั้งแบบบิตและแบบ

ไบต์

8031 มีสถาปัตยกรรมภายในดังรูป 2.1



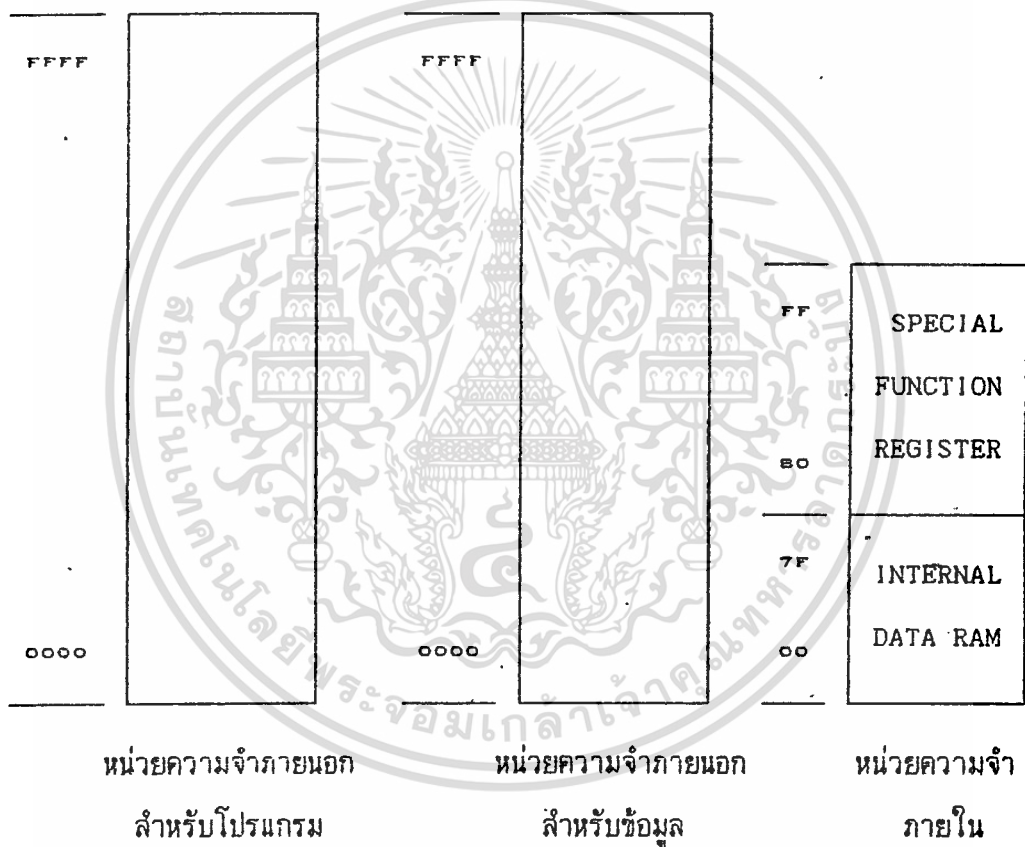
รูปที่ 2.1
สถาปัตยกรรมภายในของ 8031

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1 การจัดการหน่วยความจำ

8031 มีหน่วยความจำพื้นฐานดังนี้

1. หน่วยความจำภายนอกสำหรับโปรแกรม 64 กิโลไบต์
2. หน่วยความจำภายนอกสำหรับข้อมูล 64 กิโลไบต์
3. หน่วยความจำภายในแบบ RAM 256 ไบต์



รูปที่ 2.2
การจัดการหน่วยความจำใน 8031

ในส่วนหน่วยความจำภายในจะแบ่งเป็น 2 ส่วนหลัก ดังรูป 2.2

- แอดเดรส H00 ถึง H7F จะเป็นหน่วยความจำปกติ (RAM) สามารถเก็บข้อมูลโดยการอ้างแอดเดรส 8 บิต และในช่วง BANK จะเป็นที่ยกเก็บค่ารีจิสเตอร์ R (RAM Address Register) โดยสามารถกำหนดให้อยู่ในช่วง BANK 0, 1, 2 หรือ 3 ได้โดยเซตบิต RS1, RS0 ใน PSW รีจิสเตอร์
- แอดเดรส H80 ถึง HFF ใช้เป็นที่เก็บรีจิสเตอร์ฟังก์ชันพิเศษ (Special Function Register) การจัดแอดเดรสเหล่านี้เป็นดังรูปที่ 2.3

b.) Hardware Register BH Addresses.

Direct Byte Address	BH Addresses								Hardware Register Symbol
	(MSB)				(LSB)				
0FFH									
0F0H	F7	F6	F5	F4	F3	F2	F1	F0	S
0E0H	E7	E6	E5	E4	E3	E2	E1	E0	ACC
0C0H	D7	D6	D5	D4	D3	D2	D1	D0	PSW
0B0H	—	—	—	BC	B9	BA	B9	B2	IP
0B0H	B7	B6	B5	B4	B3	B2	B1	B0	P3
0A0H	AF	—	—	AC	AB	AA	A9	A8	IE
0A0H	A7	A6	A5	A4	A3	A2	A1	A0	P2
08H	0F	0E	0D	0C	0B	0A	09	08	SCON
08H	07	06	05	04	03	02	01	00	P1
08H	0F	0E	0D	0C	0B	0A	09	08	TCON
08H	07	06	05	04	03	02	01	00	P0

รูปที่ 2.3

รีจิสเตอร์ฟังก์ชันพิเศษ

นอกจากจะ เก็บข้อมูลโดยอ้างแอดเดรสไบท์ในหน่วยความจำได้แล้ว เรายังสามารถเก็บข้อมูลเป็นบิตได้ โดยอ้างจากแอดเดรสบิต โดยใช้คำสั่ง "MOVC" รูปที่ 2.4

RAM Byte	(MSB)				(LSB)											
7FH																
2FH									7F	7E	7D	7C	7B	7A	79	78
2EH									77	76	75	74	73	72	71	70
2DH									6F	6E	6D	6C	6B	6A	69	68
2CH									67	66	65	64	63	62	61	60
2BH									5F	5E	5D	5C	5B	5A	59	58
2AH									57	56	55	54	53	52	51	50
29H									4F	4E	4D	4C	4B	4A	49	48
28H									47	46	45	44	43	42	41	40
27H									3F	3E	3D	3C	3B	3A	39	38
26H									37	36	35	34	33	32	31	30
25H									2F	2E	2D	2C	2B	2A	29	28
24H									27	26	25	24	23	22	21	20
23H									1F	1E	1D	1C	1B	1A	19	18
22H									17	16	15	14	13	12	11	10
21H									0F	0E	0D	0C	0B	0A	09	08
20H	07	06	05	04	03	02	01	00								
1FH	Bank 3															
18H	Bank 3															
17H	Bank 3															
10H	Bank 2															
0FH	Bank 2															
08H	Bank 1															
07H	Bank 1															
00	Bank 0															

a.) RAM Bit Addresses.

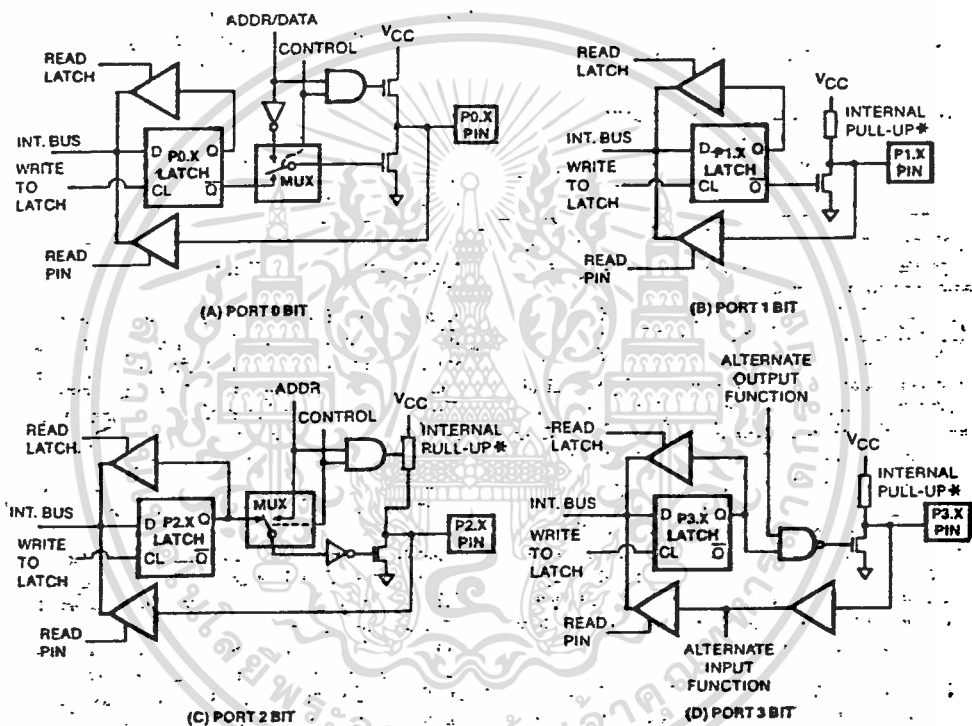
รูปที่ 2.4

การเก็บข้อมูลเป็นบิตโดยอ้างจากแอดเดรสบิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 โครงสร้างพอร์ทและการทำงาน

พอร์ทของ 8031 มี 4 พอร์ทรวม 32 บิต พอร์ททั้ง 4 เป็นแบบ 2 ทิศทาง (Bidirectional) พร้อมทั้งสามารถหน่วงค่าไว้ (Latch) โดยมีโครงสร้างภายในแต่ละบิต ดังรูปที่ 2.5



รูปที่ 2.5

โครงสร้างภายในแต่ละบิตของพอร์ท

จากรูป จะเห็นว่า มีตัวขับเอาท์พุทและบัฟเฟอร์อินพุท เนื่องจากโครงสร้าง พอร์ต 1, 2, และ 3 ได้ติดตัวความต้านทาน pull up ไว้ ดังนั้น เวลาจะอ่านค่าจาก พอร์ต ต้องป้อนค่า "1" (high) ก่อน แล้วจึงอ่านค่าที่พอร์ตเข้ามา ส่วนในพอร์ต 0 สามารถอ่านค่าได้เลย ในทางปฏิบัติพอร์ต 0 (P0) กับ P2 จะใช้ติดต่อหน่วยความ จำภายนอก โดยใช้ P0 รับข้อมูลด้วยในตัว ส่วน P1 เป็นพอร์ตว่าง ใช้ในกรณีต้องการ ติดต่ออุปกรณ์เพิ่มเติม และใน P3 จะเป็นพอร์ตควบคุม มีฟังก์ชันการทำงานดังนี้

ขาพอร์ต	ฟังก์ชันการทำงาน
P3.0	RXD (พอร์ตรับข้อมูลอนุกรม)
P3.1	TXD (พอร์ตส่งข้อมูลอนุกรม)
P3.2	INT0 (การใช้อินเทอร์รัพท์ภายนอกตัวที่ 1)
P3.3	INT1 (การใช้อินเทอร์รัพท์ภายนอกตัวที่ 2)
P3.4	T0 (Timer/Counter 0 รับสัญญาณภายนอก)
P3.5	T1 (Timer/Counter 1 รับสัญญาณภายนอก)
P3.6	WR (สไตรบการเขียนหน่วยความจำภายนอก)
P3.7	RD (สไตรบการอ่านหน่วยความจำภายนอก)

2.3 ตัวจับเวลา/ตัวนับ

ใน 8031 มีตัวจับเวลา/ตัวนับ 16 บิต 2 ตัว เราสามารถกำหนดฟังก์ชัน การทำงานได้ที่รีจิสเตอร์ TMOD ดังตารางที่ 1

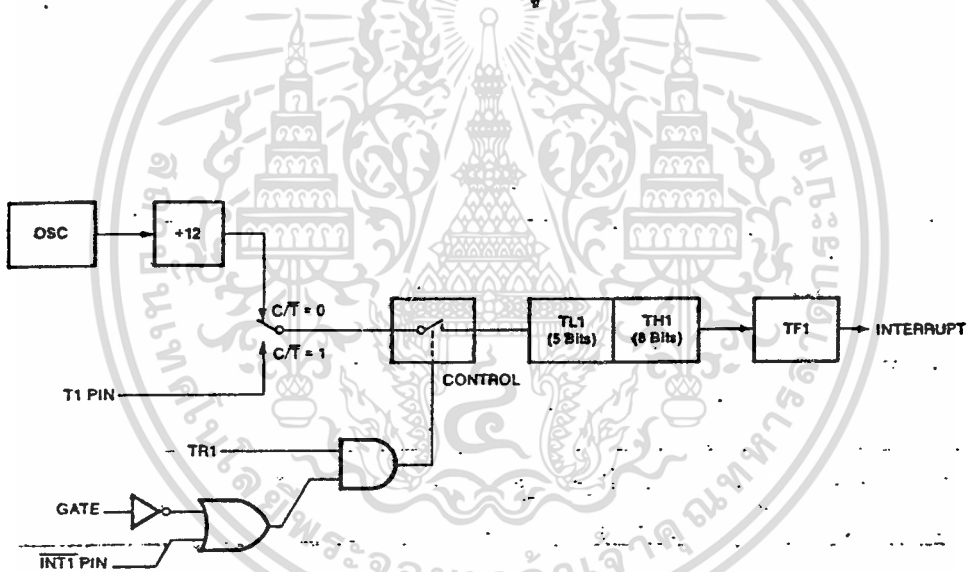
- ฟังก์ชันการทำงานจับเวลา รีจิสเตอร์จะเพิ่มค่าขึ้นทุก ๆ วัฏจักรแมชชีน (Machine Cycle) โดยใน 1 วัฏจักรแมชชีน จะกินเวลาเป็น 1/12 ของความถี่ ของออสซิลเลเตอร์ (Oscillator)

- ฟังก์ชันการทำงานเป็นตัวนับ รีจิสเตอร์จะเพิ่มค่าทุกครั้ง ที่ตรวจพบ การเปลี่ยนแปลงจาก "1" เป็น "0" ที่ขา T0 หรือ T1 สัญญาณภายนอก จะถูกลุ่ม

(Sampling) ระหว่างช่วง SSP2 ของทิวักเจอร์แมชชีน ถ้าค่าที่ได้เป็น "0" และค่าที่วัดได้ในครั้งที่แล้วเป็น "1" รีจิสเตอร์จะเพิ่มค่าอีกหนึ่ง ดังนั้นความถี่สูงสุดของสัญญาณที่จะสามารถวัดได้เป็น $1/24$ ของความถี่ออสซิลเลเตอร์

ในการทำงานทั้งตัวจับเวลา และ ตัวนับ สามารถกำหนดโหมดทำงานได้ 4 โหมด ดังนี้

โหมด ๑ รีจิสเตอร์ที่นับจะมี 2 ตัว ตัวแรกคือ TL มี 5 บิต ตัวที่ 2 คือ TH มี 8 บิต การนับโดยรวมจึงเป็น 13 บิต เมื่อนับเพิ่มค่าจนเป็น "1" ทุกบิตแล้ว การนับครั้งต่อไปจะรีเซตค่ารีจิสเตอร์เป็น "0" หมด พร้อมทั้งเกิดโอเวอร์โฟลว์ (Overflow) ไปปรับค่าแฟล็ก TF1 เป็น "1" การเริ่มจับเวลา/นับ จะมีขึ้นเมื่อบิต TR = 1 และ GATE = ๑ หรือ INT1 = 1 ดังรูปที่ 2.6

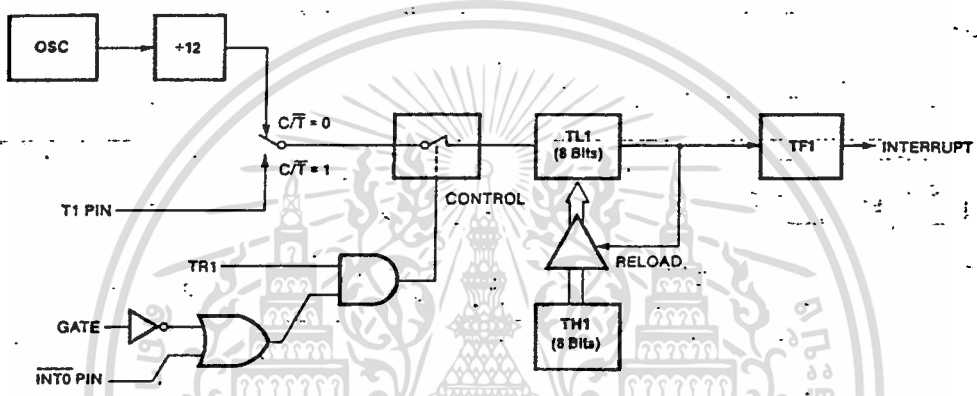


รูปที่ 2.6

การทำงานในโหมด ๑

โหมด 1 ทำงานเหมือนโหมด 0 ต่างกันที่ รีจิสเตอร์ตัวจับเวลา/นับ มีขนาด 16 บิต

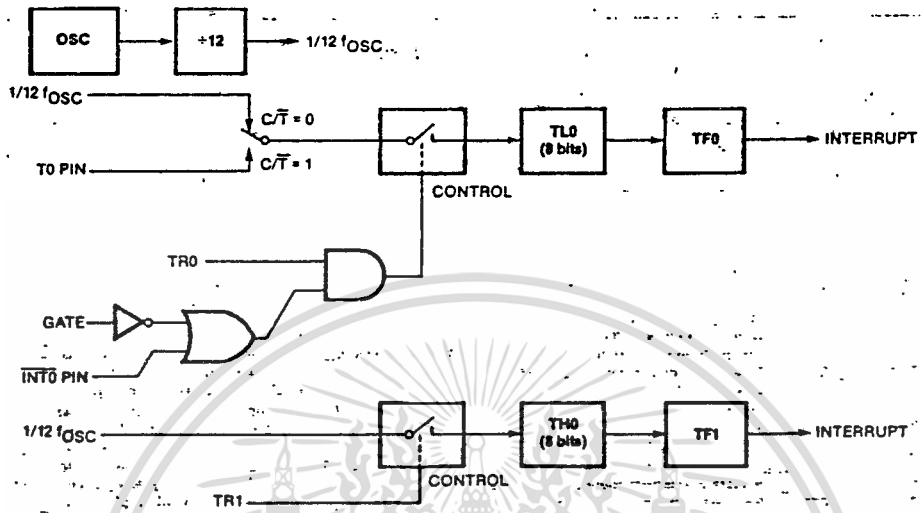
โหมด 2 ทำงานโดยเซตค่าที่ TH 8 บิต และนับค่าที่ TL 8 บิต จากนั้นทุกครั้งที่เกิดโอเวอร์โฟลว์ ที่บิต TF จะเป็น "1" และ จะมีการโหลดค่าจาก TH ไปยัง TL โดยอัตโนมัติ ดังรูปที่ 2.7



รูปที่ 2.7

การทำงานในโหมด 2

โหมด 3 จะเป็นการกำหนดให้ TL0 8 บิต เป็นตัวจับเวลา/นับ แล้วส่งสัญญาณไปเซตบิต TF0 ในขณะที่เดียวกัน TH0 จะถูกกำหนดให้ทำหน้าที่จับเวลา และส่งสัญญาณโอเวอร์โฟลว์ไปเซตบิต TF1 ดังรูปที่ 2.8



รูปที่ 2.8
การทำงานในโหมด 3

ในการรับส่งข้อมูลแบบอนุกรมโหมด 1,3 จะอาศัยการทำงานของ Timer 1 โดยอัตโนมัติ ดังนั้นในช่วงนี้จะใช้ Timer 1 ไม่ได้ และต้องเซตค่า EX1 ในรีจิสเตอร์ IE ให้ไม่ทำงานด้วย (Disabled)

2.4 การรับส่งข้อมูลแบบอนุกรม

สามารถรับและส่งได้ในเวลาเดียวกัน (Full Duplex) ในขณะที่รับ

ข้อมูลจะมีบัฟเฟอร์ช่วยทำให้สามารถรับข้อมูลในไบท์ที่สองได้ โดยที่ไบท์แรกยังไม่ถูกอ่าน อย่างไรก็ตามเมื่อรับข้อมูลในไบท์ที่สองครบแล้ว ข้อมูลในไบท์แรกจะหายไปในการรับส่ง ข้อมูลอนุกรมจะทำที่รีจิสเตอร์ SBUF ซึ่งใช้เป็นบัฟเฟอร์ โดยการเขียนข้อมูลไปที่ SBUF ข้อมูลนั้นจะถูกส่งออกไปโดยอัตโนมัติ ในการรับข้อมูล ข้อมูลที่ถูกส่งมา จะมาพักที่ตัวรีจิสเตอร์ SBUF นี้ เราจึงอ่านค่าข้อมูลได้จาก SBUF นี้เอง SBUF ในการส่งและรับ จะเป็นคนละตัวกัน โดย CPU จะรู้เอง

ในการรับส่งข้อมูลแบบอนุกรม สามารถกำหนดการทำงานได้ 4 โหมด ที่บิต SM0, SM1 ในรีจิสเตอร์ SCON ดังนี้

โหมด 0 ข้อมูลอนุกรมจะรับส่งผ่านขา RXD, TXD ตามลำดับ โดยมีความเร็วในการรับส่งเป็น 1/12 ของความถี่ออสซิลเลเตอร์ มีหน่วยเป็นบิตต่อวินาที (Baud Rate) ข้อมูลในแต่ละไบท์จะมี 8 บิต

โหมด 1 ข้อมูลจะเป็นแบบ 10 บิต มีบิตเริ่มต้น (Start Bit) บิตข้อมูล 8 บิต และบิตสุดท้าย (Stop Bit) และสามารถปรับความเร็วได้

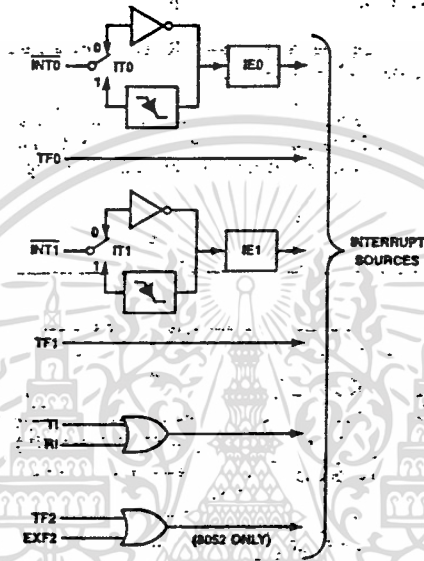
โหมด 2 ข้อมูลเป็นแบบ 11 บิต คือ บิตเริ่มต้น บิตข้อมูล 8 บิต บิตที่กำหนดค่าได้ และบิตสุดท้าย ในบิตที่กำหนดค่าได้นี้ทำได้โดยกำหนดที่บิต TB8 ของ SCON ในการส่งและการรับ บิตนี้จะถูกนำไปเก็บที่บิต RB8 ของ SCON ในทางปฏิบัติจะนิยมใช้บิตที่กำหนดค่าได้นี้เป็นพาริตีบิต (Parity Bit) สามารถเลือกความเร็วได้เป็น 1/32 หรือ 1/64 ของความถี่ออสซิลเลเตอร์

โหมด 3 จะเหมือนโหมด 2 ทุกอย่าง ยกเว้นความเร็วสามารถปรับค่าได้

ในทุกโหมด การส่งข้อมูล จะเริ่มต้นที่มีการเขียนข้อมูลไปยัง รีจิสเตอร์ SBUF ส่วนในการรับข้อมูลในโหมด 0 จะเริ่มเมื่อสถานะ RI = 0 และ REN = 1 แต่ในโหมดอื่นจะเริ่มรับข้อมูลโดย REN = 1 และมีบิตเริ่มต้นเข้ามา

2.5 อินเทอร์รัพท์ (Interrupt)

มีการอินเทอร์รัพท์ได้ 5 วิธี ดังรูปที่ 2.9



รูปที่ 2.9

แสดงการอินเทอร์รัพท์

การอินเทอร์รัพท์จากภายนอก INT0 และ INT1 สามารถกำหนดการทำงานได้ทั้งแบบระดับ (Level) หรือแบบการเปลี่ยนระดับ (Transition) โดยเซตค่าที่บิต IT0, IT1 ใน TCON เมื่อมีการอินเทอร์รัพท์ บิต IE0, IE1 ใน TCON จะถูกเซตเป็น "1" และจะถูกเคลียร์เป็น "0" เมื่อสิ้นสุดขบวนการอินเทอร์รัพท์

การอินเทอร์รัพท์จากตัวจับเวลา/ตัวนับ TF0 และ TF1 ใน TCON จะถูก

เซ็ทเมื่อ รีจิสเตอร์ในตัวจับเวลานั้นนับถึง 255 และถูกเคลียร์โดยฮาร์ดแวร์ เมื่อมีการชี้ไปยังตำแหน่งโปรแกรมตอบสนองอินเทอร์รัพท์แล้ว

การอินเทอร์รัพท์ทางพอร์ทอนุกรม จะถูกอินเทอร์รัพท์ จากการรับข้อมูลหรือจากการส่งข้อมูล บิต R1 และ T1 จะถูกเซ็ทตามลำดับ และจะไม่ถูกเคลียร์โดยทางฮาร์ดแวร์ เพื่อเราจะได้สามารถตรวจเช็คการรับส่งข้อมูลในครั้งที่แล้วได้ บิต R1 และ T1 จะถูกเซ็ทตามลำดับ และจะไม่ถูกเคลียร์โดยทางฮาร์ดแวร์ เพื่อเราจะได้ตรวจเช็คการรับส่งข้อมูลในครั้งที่แล้วได้ บิต R1 หรือ T1 จะถูกเคลียร์ โดยทางซอฟต์แวร์ทุกครั้ง ก่อนจะมีการรับส่งข้อมูล

เราสามารถกำหนดลำดับความสำคัญในการอินเทอร์รัพท์ได้ โดยทางซอฟต์แวร์ที่รีจิสเตอร์ IP ดังรูป และเลือกใช้อินเทอร์รัพท์ได้โดยกำหนดที่รีจิสเตอร์ IE ดังรูปที่ 2.10

(MSB)							(LSB)								
EA	X	ET2	ES	ET1	EX1	ET0	EX0	X	X	PT2	PS	PT1	PX1	PT0	PX0
Symbol	Position	Function													
EA	IE.7	disables all interrupts. If EA = 0, no interrupt will be acknowledged. If EA = 1, each interrupt source is individually enabled or disabled by setting or clearing its enable bit.													
—	IE.6	reserved													
ET2	IE.5	enables or disables the Timer 2 overflow or capture interrupt. If ET2 = 0, the Timer 2 interrupt is disabled.													
ES	IE.4	enables or disables the Serial Port interrupt. If ES = 0, the Serial Port interrupt is disabled.													
ET1	IE.3	enables or disables the Timer 1 Overflow interrupt. If ET1 = 0, the Timer 1 interrupt is disabled.													
EX1	IE.2	enables or disables External Interrupt 1. If EX1 = 0, External Interrupt 1 is disabled.													
ET0	IE.1	enables or disables the Timer 0 Overflow interrupt. If ET0 = 0, the Timer 0 interrupt is disabled.													
EX0	IE.0	enables or disables External Interrupt 0. If EX0 = 0, External Interrupt 0 is disabled.													
—	IP.7	reserved													
—	IP.6	reserved													
PT2	IP.5	defines the Timer 2 interrupt priority level. PT2 = 1 programs it to the higher priority level.													
PS	IP.4	defines the Serial Port interrupt priority level. PS = 1 programs it to the higher priority level.													
PT1	IP.3	defines the Timer 1 interrupt priority level. PT1 = 1 programs it to the higher priority level.													
PX1	IP.2	defines the External Interrupt 1 priority level. PX1 = 1 programs it to the higher priority level.													
PT0	IP.1	defines the Timer 0 interrupt priority level. PT0 = 1 programs it to the higher priority level.													
PX0	IP.0	defines the External Interrupt 0 priority level. PX0 = 1 programs it to the higher priority level.													

รูปที่ 2.10

การเลือกใช้และการจัดลำดับความสำคัญในการอินเทอร์รัพท์

ตารางที่ 1 การกำหนดค่าบิตในรีจิสเตอร์ TMOD

Timer 1				Timer 0			
Gate	C/T	M1	M0	Gate	C/T	M1	M0

(MSB)

(LSB)

- Gate ถ้ามีค่าเป็น 0 จะเป็นตัวที่ทำให้ Timer หรือ Counter ทำงานได้
- C/T เป็นตัวเลือกว่าจะใช้งานเป็น Timer หรือ Counter โดย 0 เป็น Timer และ 1 เป็น Counter
- M1, M2 เป็นตัวเลือกโหมดการทำงาน
- M1,M2=00 เป็นการใช้ Timer/Counter ขนาด 13 บิต
- M1,M2=01 เป็นการใช้ Timer/Counter ขนาด 16 บิต
- M1,M2=10 เป็นการใช้ Timer/Counter ขนาด 8 บิต โดยเป็นลักษณะ เรียกว่าเข้ามาโดยอัตโนมัติ (Auto-reload)
- M1,M2=11 ใน Timer 0 เป็นการใช้ Timer/Counter ขนาด 8 บิต โดย TL0 จะถูกควบคุมตามมาตรฐาน ส่วน TH0 จะถูกควบคุมจาก TR1 ใน Timer 1 จะหยุดใ้

ตารางที่ 2 การกำหนดค่าบิตในรีจิสเตอร์ IE

EA	X	X	ES	ET1	EX1	ET0	EX0
----	---	---	----	-----	-----	-----	-----

(MSB)

(LSB)

- EA เป็นการควบคุมการรับ INT ของทั้งระบบ ถ้า EA = 0 จะไม่รับ INT ถ้า EA = 1 การรับ INT จะขึ้นอยู่กับความพร้อมที่จะรับ INT ของแต่ละตัว ซึ่งจะเปิดสลับต่อกัน
- ES เป็นตัวควบคุมการรับ INT ของ Serial Port
- ET1 เป็นตัวควบคุมการรับ INT ของ Timer 1
- EX1 เป็นตัวควบคุมการรับ INT ของ INT1
- ET0 เป็นตัวควบคุมการรับ INT ของ Timer 0
- EX0 เป็นตัวควบคุมการรับ INT ของ INT0
- X สำรอง

ตารางที่ 3 การกำหนดค่าบิตในรีจิสเตอร์ TCON

TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0
-----	-----	-----	-----	-----	-----	-----	-----

(MSB)

(LSB)

- TF1 เป็นแฟล็กโอเวอร์โฟลว์ของ Timer/Counter 1 การเซตและเคลียร์ จะกระทำทางฮาร์ดแวร์
- TR1 เป็นตัวควบคุมตัว Timer 1 ให้ทำงานหรือหยุดทำงาน การเซตและเคลียร์ จะกระทำทางซอฟต์แวร์
- TF0 เป็นแฟล็กโอเวอร์โฟลว์ของ Timer/Counter 0
- TR0 เป็นตัวควบคุมตัว Timer 0
- IE1 เป็นตัวตรวจว่ามีการรับสัญญาณ INT1 หรือไม่
- IT1 เป็นบิตที่กำหนดการรับสัญญาณ INT1 ว่า จะรับช่วงขาลง หรือช่วงสถานะ เป็น 0 ของสัญญาณ INT1

- IE0 เป็นตัวตรวจว่ามีการรับสัญญาณ INTO หรือไม่
- IT0 เป็นบิตที่กำหนดการรับสัญญาณ INTO ว่า จะรับช่วงกลาง หรือช่วงสถานะ เป็น 0 ของสัญญาณ INTO

ตารางที่ 4 การกำหนดค่าบิตในรีจิสเตอร์ PSW

CY	AC	F0	RS1	RS0	OV	X	P
----	----	----	-----	-----	----	---	---

(MSB)

(LSB)

- CY เป็นแครี่แฟล็ก (Carry Flag)
- AC เป็นออกซิลลารีแฟล็ก (Auxillary Flag) สำหรับการทํางานแบบบีซีดี (BCD)
- F0 เป็นแฟล็กศูนย์ (Flag 0)
- RS1 เป็นบิตที่ใช้เลือกกริจิสเตอร์แบงค์ (Register Bank) สามารถตั้งและลบด้วยซอฟต์แวร์
- RS0 แสดงการทํางานของกริจิสเตอร์แบงค์
- OV เป็นแฟล็กโอเวอร์โฟลว์ (Overflow Flag)
- X สำรอง
- P เป็นแฟล็กพาริตี (Parity Flag) ตั้งและลบโดยฮาร์ดแวร์

- หมายเหตุ** รูปแบบการตั้งการทํางานของกริจิสเตอร์แบงค์ โดยตั้ง (RS0, RS1)
- (0, 0) เลือกการทํางานช่วง (00H - 07H)
 - (0, 1) เลือกการทํางานช่วง (08H - 0FH)
 - (1, 0) เลือกการทํางานช่วง (10H - 17H)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(1,1) เลือกการทำงานช่วง (18H - 1FH)

ตารางที่ 5 การกำหนดค่าบิตในรีจิสเตอร์ IP

X	X	X	PS	PT1	PX1	PT0	PX0
---	---	---	----	-----	-----	-----	-----

(MSB)

(LSB)

- PS เป็นการทำให้ INT ของ Serial Port มีความสำคัญกว่าอันอื่น
- PT1 เป็นการทำให้ INT ของ Timer 1 มีความสำคัญกว่าอันอื่น
- PX1 เป็นการทำให้ INT ของ INT1 มีความสำคัญกว่าอันอื่น
- PT0 เป็นการทำให้ INT ของ Timer 0 มีความสำคัญกว่าอันอื่น
- PX0 เป็นการทำให้ INT ของ INT0 มีความสำคัญกว่าอันอื่น
- X สำรอง

ตารางที่ 6 การกำหนดค่าบิตในรีจิสเตอร์ SCON

SM0	SM1	SM2	REN	TB8	RB8	T1	R1
-----	-----	-----	-----	-----	-----	----	----

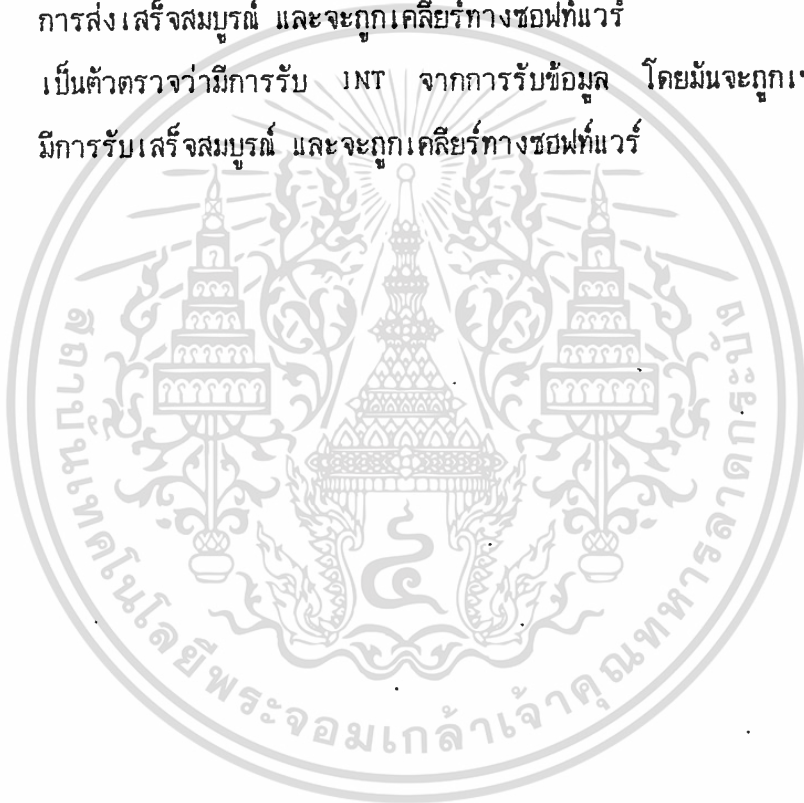
(MSB)

(LSB)

- SM0, SM1 เป็นตัวเลือกโหมดควบคุม
- SM2 เป็นตัวทำให้มีการติดต่อแบบมัลติโพรเซสเซอร์ (Multiprocessor) ได้ในโหมด 1, 2, 3 ถ้า SM2 ถูกเซตเป็น 1 จะทำให้ R1 ไม่สามารถทำงานใน

โหมด ๑ ได้ SM2 คาร์เป็น 1

- REN เป็นตัวทำให้สามารถรับส่งแบบ Serial ได้ ถ้าเซทจะทำให้รับส่งได้ ถ้า
เคิลียร์จะไม่สามารถส่งข้อมูลได้
- TB8 เป็นบิทที่ 9 ของข้อมูลที่จะส่งไปในโหมด 2 หรือ 3
- RB8 เป็นบิทที่ 9 ของข้อมูลที่จะส่งไปในโหมด 2 หรือ 3 แต่ในโหมด 1 จะ
เป็นค่า stop บิท และในโหมด ๑ จะไม่ถูกใช้
- T1 เป็นตัวตรวจว่ามีการรับ INT จากการส่งข้อมูล โดยมันจะถูกเซท เมื่อ
การส่งเสร็จสมบูรณ์ และจะถูกเคลียร์ทางซอฟต์แวร์
- R1 เป็นตัวตรวจว่ามีการรับ INT จากการรับข้อมูล โดยมันจะถูกเซท เมื่อ
มีการรับเสร็จสมบูรณ์ และจะถูกเคลียร์ทางซอฟต์แวร์



บทที่ 3

หลักการงานและการออกแบบวงจร

3.1 หลักการงาน

การควบคุมอุณหภูมิ ในขบวนการนี้ จัดเป็นรูปการควบคุมแบบดิจิตอลโดยตรง โดยมีส่วนประกอบที่สำคัญ 2 ส่วนใหญ่ ๆ คือ

1. ฮาร์ดแวร์ (hardware)

ประกอบด้วย ส่วนของการวัดค่าอุณหภูมิ จากขบวนการ และแปลงเป็นค่าแรงดันไฟฟ้า การแปลงค่าแรงดันไฟฟ้าเป็นสัญญาณดิจิตอล ส่วนของตัวควบคุม และส่วนที่จะแปลงสัญญาณดิจิตอลไปควบคุมขบวนการ

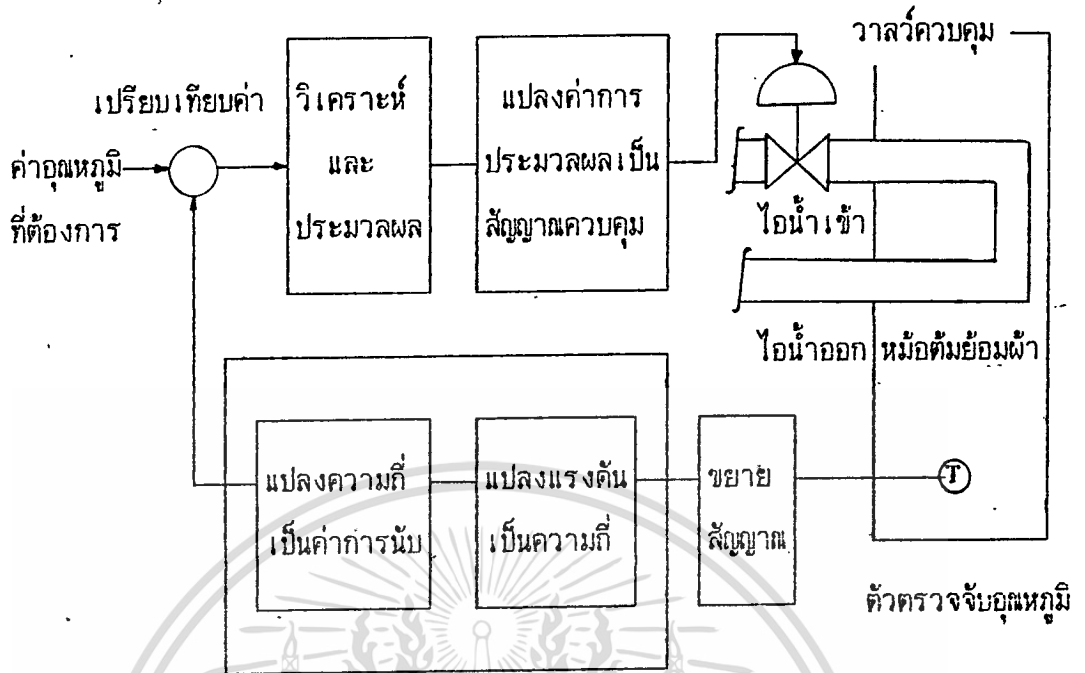
2. ซอฟต์แวร์ (software)

ประกอบด้วย โปรแกรมการทำงานของตัวควบคุม ซึ่งจะกำหนดการทำงานของ การควบคุม

ฮาร์ดแวร์ (hardware)

ฮาร์ดแวร์ มีโครงสร้างส่วนประกอบต่าง ๆ เป็นไปดังรูปที่ 3.1

จากขบวนการ ค่าอุณหภูมิจะถูกแปลงเป็นระดับแรงดันไฟฟ้า โดยตัวตรวจจับอุณหภูมิ ซึ่งในโครงการนี้ ใช้ตัวเทอร์โมคัปเปิลแบบซีเอ (CA type) แรงดันไฟฟ้าที่ได้ จะผ่านการขยายให้อัตราการเปลี่ยนของแรงดันไฟฟ้า ต่อ การเปลี่ยนแปลงอุณหภูมิ อยู่ในระดับ 10 มิลลิโวลต์ ต่อ 1 องศาเซลเซียส จากนั้น ค่าแรงดันไฟฟ้าจะถูกเปลี่ยนให้เป็นความถี่ โดยใช้อัตราการเปลี่ยนแปลงของความถี่ ต่อ การเปลี่ยนแปลงของระดับแรงดันไฟฟ้าเป็น 1 กิโลเฮิรตซ์ ต่อ โวลต์ ความถี่ที่ได้จะใช้นำไปเข้าวงจรนับ เพื่อแปลงค่าความถี่เป็นข้อมูล เข้าไปยังตัวควบคุม



แปลงอนาลอกเป็นดิจิทัล

รูปที่ 3.1

โครงสร้างส่วนประกอบต่าง ๆ ของยาร์ดแวร์

ภายในตัวควบคุม จะมีหน่วยความจำในการเก็บค่าอุณหภูมิที่ต้องการ การทำงานของตัวควบคุม จะเริ่มจากการอ่านค่าอุณหภูมิของขบวนการจากวงจรนับ แล้วทำการเปรียบเทียบค่ากับข้อมูลที่ตั้งไว้ ค่าที่ได้จากการเปรียบเทียบ ก็จะถูกประมวลผลในหน่วยประมวลผลกลาง หลังจากนั้น ผลจากการประมวลผลก็จะถูกแปลงกลับเป็นสัญญาณไฟฟ้า เพื่อนำไปควบคุมการทำงานของวาล์วควบคุม ในการควบคุมการไหลของไอน้ำเข้าไปยังหม้อต้ม เพื่อควบคุมอุณหภูมิต่อไป

ซอฟต์แวร์ (software)

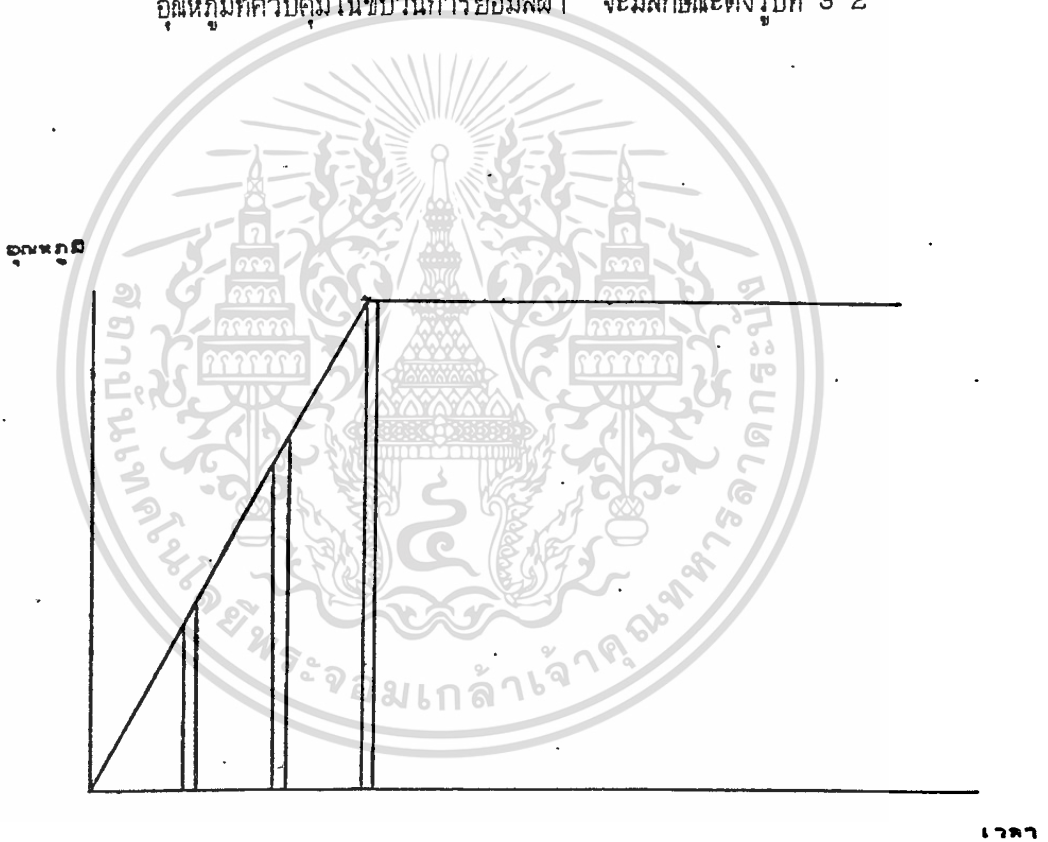
ส่วนนี้เป็นส่วนที่เปรียบเสมือนหัวใจในการทำงานของตัวควบคุม การเปรียบเทียบ

เทียบค่า และการประมวลผลในตัวควบคุม ก็จะถูกกำหนดในส่วนของซอฟต์แวร์

ซอฟต์แวร์ ประกอบไปด้วย โปรแกรมการทำงานที่ควบคุมการทำงานของตัวควบคุมเอง คือ การรับค่าจากผู้ใช้ การแสดงผลการควบคุม และการโปรแกรมการทำงานในการควบคุมอุณหภูมิในขบวนการ โดยใช้การควบคุมแบบเปิด-ปิด (on-off control) ในการควบคุม

การควบคุมอุณหภูมิในขบวนการย้อมสีผ้า

อุณหภูมิที่ควบคุมในขบวนการย้อมสีผ้า จะมีลักษณะดังรูปที่ 3.2



รูปที่ 3.2

อุณหภูมิที่ควบคุมในขบวนการย้อมสีผ้า

การควบคุมอุณหภูมิจะถูกแบ่งออกเป็นช่วง ๆ โดยอุณหภูมิจะถูกควบคุมให้มีค่าเป็นไปตามที่กำหนดในแต่ละช่วงเวลา เพื่อให้ได้คุณสมบัติของการย้อมสีผ้าตามที่ต้องการ และหลังจากการจบในแต่ละช่วงการควบคุม จะมีการหยุดการทำงาน เพื่อเติมสารเคมีที่เหมาะสมลงไปในช่วงการ

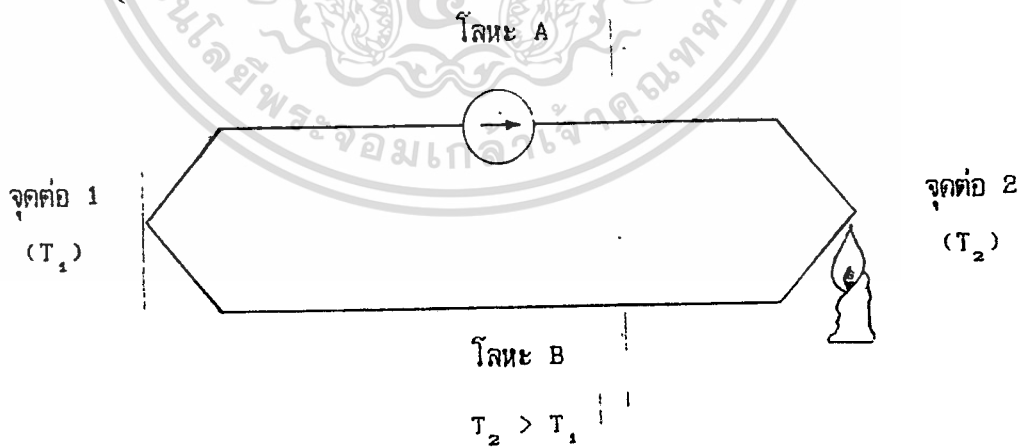
3.2 การออกแบบวงจร

ส่วนวัดอุณหภูมิและแปลงเป็นสัญญาณไฟฟ้า

ประกอบด้วย ตัววัดอุณหภูมิ วงจรขยาย วงจรชดเชยอุณหภูมิโดยใช้แรงดันไฟฟ้า และวงจรรวมสัญญาณ

1. ตัววัดอุณหภูมิ (TEMPERATURE TRANSDUCER)

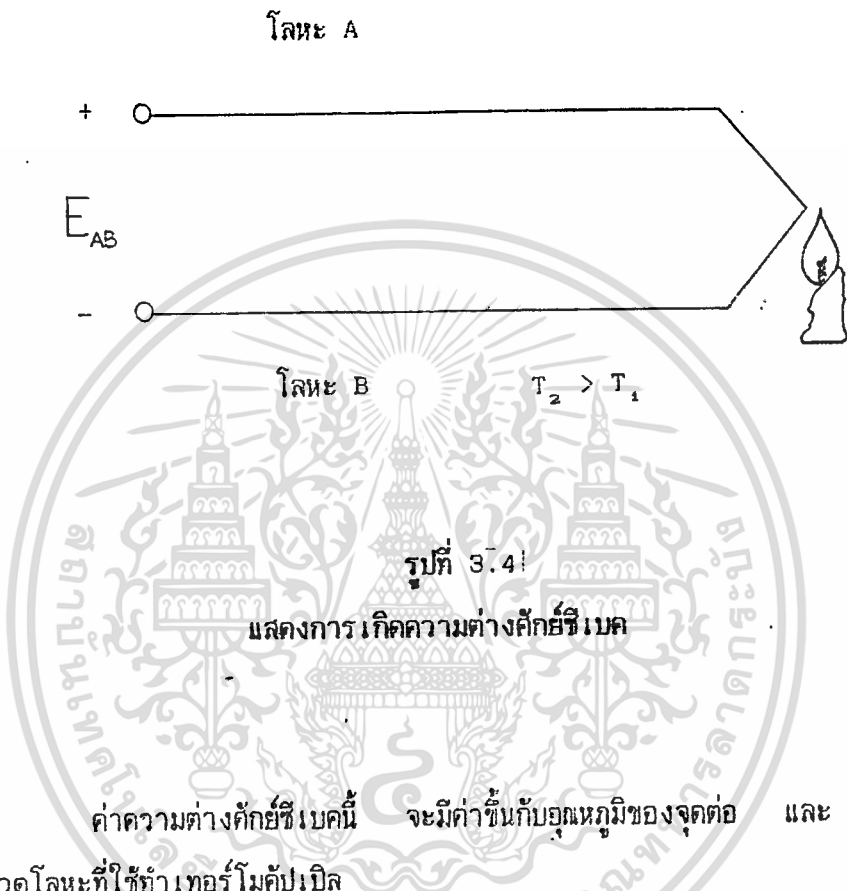
ใช้เทอร์มิสเตอร์เป็นตัวตรวจวัดอุณหภูมิของขบวนการ และแปลงเป็นแรงดันไฟฟ้า หลักการของเทอร์มิสเตอร์มีอยู่ว่า เมื่อนำเส้นลวดโลหะต่างชนิดกัน 2 เส้น มาเชื่อมเข้าด้วยกันที่ปลายทั้งสองข้าง และให้ความร้อนที่ปลายด้านหนึ่ง จะเกิดกระแสไหลในเส้นลวดทั้งสองเส้นนั้น ดังรูปที่ 3.3



รูปที่ 3.3

แสดงหลักการทำงานของเทอร์มิสเตอร์

และจากรูปที่ 3.3 ถ้าเราเปิดวงจรที่จุดต่อ 1 ก็เกิดค่าความต่างศักย์ E_{AB} ขึ้น เรียกว่า ความต่างศักย์ซีเบค (Seebeck voltage) ดังรูปที่ 3.4



รูปที่ 3.4
แสดงการเกิดความต่างศักย์ซีเบค

ค่าความต่างศักย์ซีเบคนี้ จะมีค่าขึ้นกับอุณหภูมิของจุดต่อ และ ชนิดของเส้นลวดโลหะที่ใช้ทำเทอร์โมคัปเปิล

ในโครงการนี้ใช้เทอร์โมคัปเปิลแบบซีเอ ซึ่งมีค่าสัมประสิทธิ์ซีเบค เท่ากับ 40.7 ไมโครโวลท์ ต่อ องศาเซลเซียส โดย E_{AB} ที่วัดได้จะเป็นไปตามสมการ

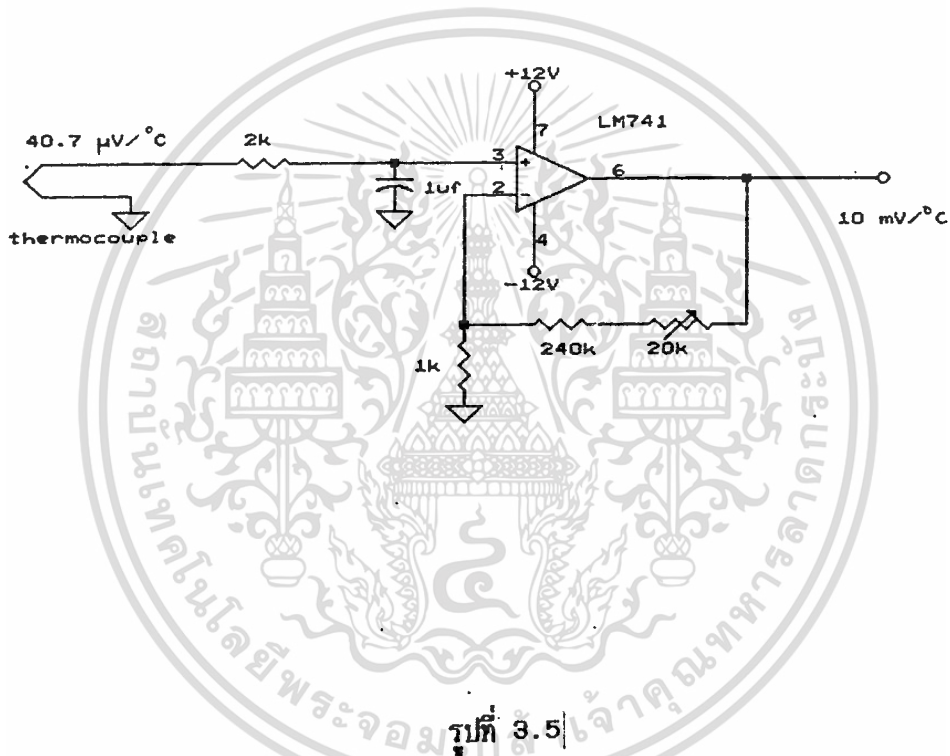
$$E_{AB} = \alpha(T_2 - T_1)$$

- เมื่อ α : สัมประสิทธิ์ซีเบค
- T_1 : อุณหภูมิที่ปลายข้างหนึ่ง
- T_2 : อุณหภูมิที่ปลายอีกข้างหนึ่ง

2. วงจรขยายสัญญาณ

การเปลี่ยนแปลงของระดับแรงดันจากเทอร์โมคัปเปิลมีค่าน้อยมาก (40.7 ไมโครโวลต์ ต่อ องศาเซลเซียส) จึงต้องทำการขยายสัญญาณเป็น 10 มิลลิโวลต์ ต่อ องศาเซลเซียส

วงจรขยายสัญญาณที่ใช้ เป็นแบบนอนอินเวอร์ติ้งแอมป์ (Non inverting amplifier) ดังรูปที่ 3.5



แสดงวงจรขยายรวมทั้งวงจรรองความถี่ต่ำผ่านเพื่อกรองสัญญาณรบกวน

โดยอัตราการขยายแรงดันของวงจรมีค่าเท่ากับ 10 มิลลิโวลต์ ต่อ องศาเซลเซียส หากด้วย 40.7 ไมโครโวลต์ ต่อ องศาเซลเซียส ซึ่งจะเท่ากับ 245.7 เท่า

3. วงจรชดเชยอุณหภูมิโดยใช้แรงดันไฟฟ้า

ค่าของความต่างศักย์ซีเบคจากเทอร์โมคัปเปิลนี้ เป็นค่าที่เกิดจากค่าแตกต่างกัน

ระหว่างอุณหภูมิ T_2 และ T_1 ตามปกติในการใช้เทอร์โมคัปเปิลจะต้องมีการรักษาอุณหภูมิ T_1 ไว้ที่ศูนย์องศาเซลเซียส เพื่อให้ค่าความต่างศักย์ที่วัดได้เป็นค่าที่อ้างอิงอยู่กับอุณหภูมิศูนย์องศาเซลเซียส แต่การรักษาอุณหภูมิของปลายข้างหนึ่งของเทอร์โมคัปเปิลไว้ที่ ศูนย์ องศาเซลเซียส เป็นเรื่องยุ่งยากมาก ดังนั้นเทคนิคหนึ่งที่น่ามาใช้แทน คือ การชดเชยด้วยระดับแรงดันไฟฟ้า ซึ่งทำให้ไม่ต้องรักษาอุณหภูมิของปลายข้างหนึ่งไว้ที่ ศูนย์องศาเซลเซียส

การชดเชยระดับแรงดันไฟฟ้านี้ จะทำการบวกค่าแรงดันไฟฟ้าเข้ากับค่าแรงดันที่วัดได้จากเทอร์โมคัปเปิล ซึ่งเกิดจากค่าผลต่างของอุณหภูมิขบวนการและอุณหภูมิจุดอ้างอิง โดยแรงดันที่จะบวกเพิ่มเข้าไปนี้มีค่าเท่ากับแรงดันที่เกิดจากเทอร์โมคัปเปิลที่วัดอุณหภูมิห้องเทียบกับ ศูนย์ องศาเซลเซียส นิยามได้จากสมการดังนี้

ก่อนชดเชย : $E_{AB} = e(T_2 - T_1)$

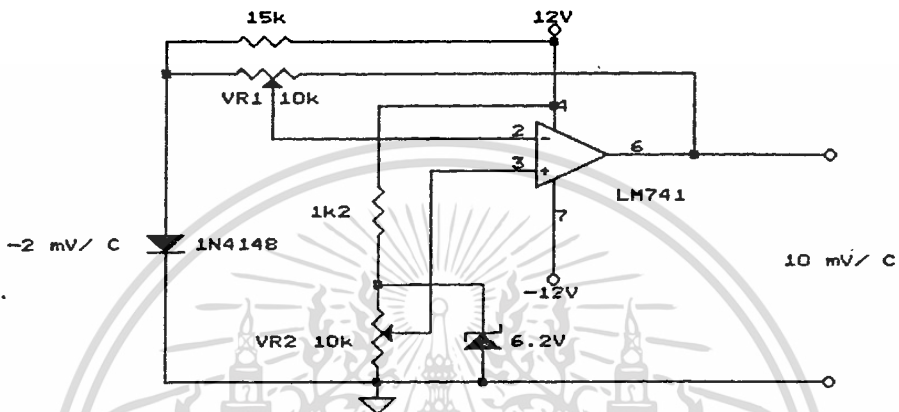
T_1 : อุณหภูมิห้อง

T_2 : อุณหภูมิของขบวนการ

หลังชดเชย : $E_{NEW} = E_{AB} + E_{COMPENSATE}$
 $e(T_2 - T_0) = e(T_2 - T_1) + eT_1$

นั่นคือ $E_{COMPENSATE} = eT_1$

วงจรที่จะใช้ชดเชยแรงดันไฟฟ้านี้ จะต้องให้ค่าแรงดันเอาท์พุทของมันแปรไปตามอุณหภูมิห้องที่เปลี่ยนไปด้วย วงจรที่ใช้ในโครงการนี้ อาศัยคุณสมบัติของ DIODE มาเป็นตัวตรวจจับอุณหภูมิห้อง ดังรูปที่ 3.6 |



รูปที่ 3.6

แสดงวงจรวัดอุณหภูมิห้องเพื่อนำไปทดสอบแรงดัน

จากคุณสมบัติของซิลิกอนไดโอดเบอร์ 4148 ซึ่งมีสัมประสิทธิ์การเปลี่ยนแปลงของแรงดันตกคร่อมต่ออุณหภูมิเท่ากับ -2 มิลลิโวลต์ ต่อ องศาเซลเซียส ขยายสัญญาณโดยใช้วงจรแบบอินเวอร์ติ้งแอมป์ (Inverting amplifier) การปรับค่าอัตราขยายของวงจรจะทำได้โดยการปรับ VR_1 ระดับแรงดันเอาต์พุตของวงจรจะเป็นไปตามสมการ

$$V = -(VR_{1/2} / VR_{1/1}) T_c$$

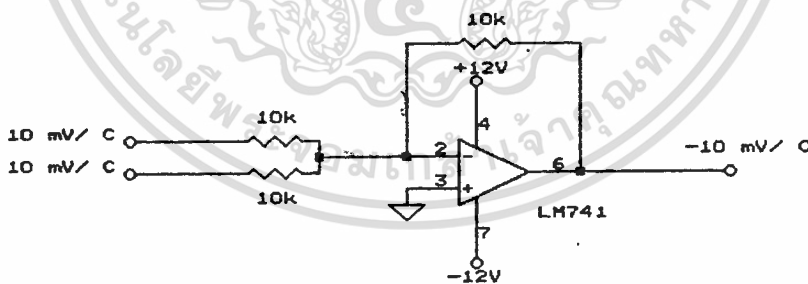
T_c : ลัมประสิทธิการ เปลี่ยนอุณหภูมิของไดโอด

ซีเนอร์ไดโอดทำหน้าที่ จ่ายแรงดันอ้างอิงแก่ขานอนอินเวอร์ตติ้งของออปแอมป์ ตัวต้านทาน VR_2 ที่ต่อขนานกับซีเนอร์ไดโอดจะเป็นตัวปรับแรงดันออฟเซตที่แรงดันเอาต์พุต ของออปแอมป์

การปรับค่า VR_1 ต้องปรับเพื่อให้ได้อัตราการเปลี่ยนแปลงแรงดันเอาต์พุตต่อการเปลี่ยนแปลงอุณหภูมิห้องเป็น 10 มิลลิโวลท์ ต่อ องศาเซลเซียส เสร็จแล้วนำไปวัดอุณหภูมิที่รู้ค่าแน่นอน เพื่อปรับ VR_2 ให้ได้แรงดันเอาต์พุตที่ต้องการ

4. วงจรรวมสัญญาณ

ค่าระดับแรงดันจากวงจรรขยายสัญญาณ และ จากวงจรถดเชยอุณหภูมิจะถูกรวมกันโดยใช้วงจรอินเวอร์ตติ้งซัมมิงแอมป์ (INVERTING SUMMING AMP) ตามรูปที่ 3.7 ค่าระดับแรงดันที่ได้จากวงจรรวมสัญญาณนี้ก็คือ ค่าแรงดันที่เปรียบเสมือนใช้ เทอร์โมคัปเปิลวัดอุณหภูมิของขบวนการ เทียบกับ อุณหภูมิศูนย์องศาเซลเซียส



รูปที่ 3.7|

แสดงวงจรรวมสัญญาณ

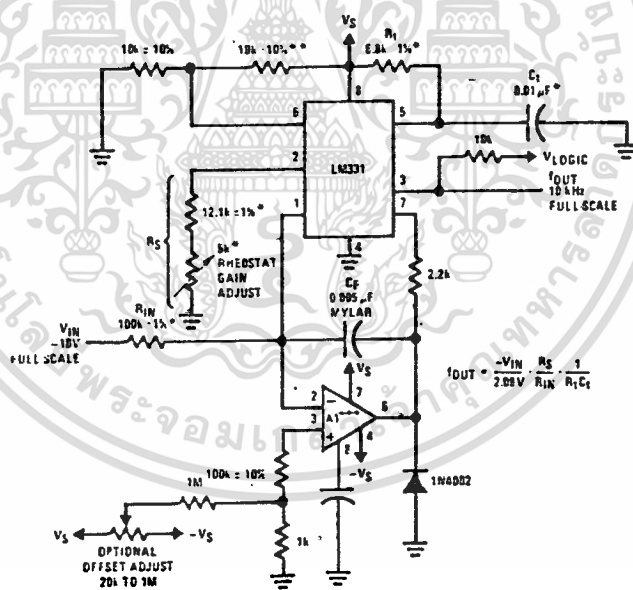
ส่วนแปลงค่าแรงดันไฟฟ้าเป็นสัญญาณดิจิทัล

ประกอบด้วยวงจรเปลี่ยนระดับแรงดันเป็นความถี่ และ วงจรนับความถี่

1. วงจรแปลงระดับแรงดันเป็นความถี่ (V to F CONVERTER)

ตามปกติในการส่งค่าของสัญญาณในรูปของแรงดันต่อเนื่อง มักจะมีปัญหาเกี่ยวกับสัญญาณรบกวน ที่ทำให้ค่าของสัญญาณผิดเพี้ยนไป วิธีการแก้ปัญหาเกี่ยวกับสัญญาณรบกวนวิธีหนึ่งคือ ใช้วงจรแปลงระดับแรงดันให้เป็นความถี่ เพราะว่าสัญญาณรบกวนจะมีผลต่อแอมพลิจูดของสัญญาณแต่ไม่มีผลต่อความถี่ของสัญญาณ

วงจรแปลงนี้จะให้เอาท์พุทออกมา เป็นคลื่นสี่เหลี่ยมที่มีความถี่เปลี่ยนไปตามระดับแรงดันอินพุท วงจรแปลงระดับแรงดันเป็นความถี่ที่ใช้ในโครงการนี้ใช้ตัววงจรรวม LM331 ซึ่งเป็นวงจรแปลงระดับแรงดันเป็นความถี่สำเร็จรูปภายในตัว ดังรูปที่ 3.8 |



- *Use stable components with low temperature coefficients. See Typical Applications section.
- **This resistor can be 5 kΩ or 10 kΩ for $V_S = 8V$ to $22V$, but must be 10 kΩ for $V_S = 4.5V$ to $8V$.
- ***Use low offset voltage and low offset current op amps for A1: recommended types LM108, LM308A, LF351B

รูปที่ 3.8 |

แสดงวงจรแปลงระดับแรงดันเป็นความถี่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ค่าความถี่เอาต์พุทของวงจรถูกจะเป็นไปตามสมการ

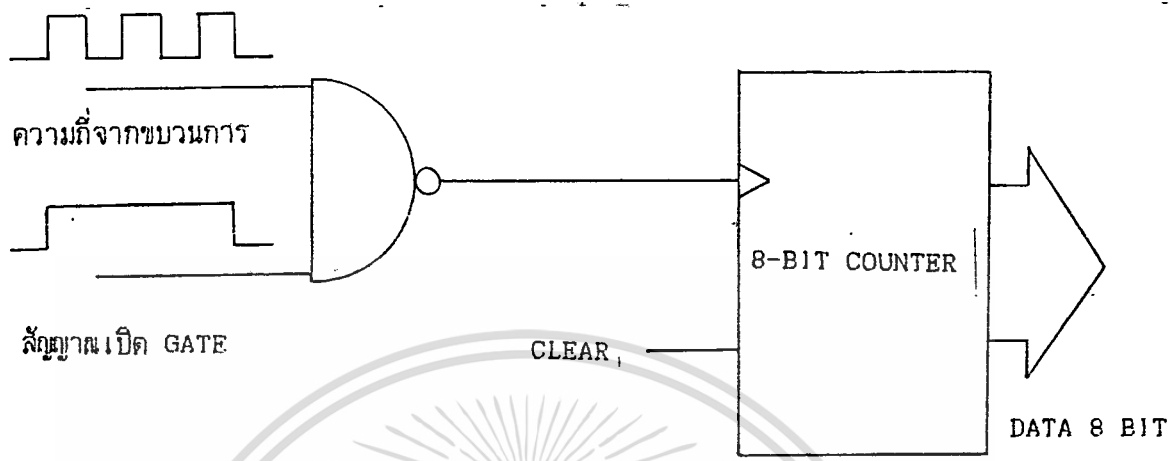
$$f_{out} = \frac{V_{in} * R_u * 1}{2.09 V * R_{in} * R_c C_c}$$

ในโครงงานนี้เรากำหนดให้อัตราการเปลี่ยนแปลงความถี่ต่อการเปลี่ยนแปลงของระดับแรงดันอินพุทมีค่าเท่ากับ 1 กิโลเฮิรตซ์ ต่อ โวลต์

2. วงจรนับความถี่ (Frequency counter)

ความถี่ที่ได้จากวงจรถ่ายระดับแรงดันเป็นความถี่จะนำมาผ่านตัวแยกสัญญาณด้วยแสง (Opto isolator) เพื่อแยกกราวด์ของวงจรถ่ายส่วนอนาล็อก และ วงจรส่วนดิจิทัล ออกจากกัน จากนั้นก็จะผ่านแอนเนกทชมิทริกเกอร์ (Nand-gate schmitt trigger) เพื่อเป็นสัญญาณไปทริกวงจรถ่ายขนาด 8 บิตเพื่อเป็นข้อมูลให้กับตัวควบคุมดิจิทัลต่อไป

ในการทำงานเพื่อรับค่าอนุพัทธ์จากขบวนการ ตัวควบคุมซึ่งใช้ไมโครคอนโทรลเลอร์ 8031 เป็นส่วนประมวลผลกลางจะกำหนดช่วงเวลาการนับโดยใช้ตัวจับเวลา (Timer) ภายใน ค่าลอจิก "1" จะถูกส่งในลักษณะของโมโนสเตเบิลออกมาตามช่วงเวลาที่กำหนดเข้าไปยังอินพุทของแอนเนกทชมิทริกเกอร์ เพื่อเปิดเกตให้สัญญาณความถี่เข้าไปทริกวงจรถ่าย เมื่อครบช่วงเวลาสัญญาณเปิดเกตก็จะมีค่าลอจิก "0" เป็นการหยุดการนับ จากนั้นตัวควบคุมก็จะอ่านค่าที่ได้จากการนับไปเป็นข้อมูลในการประมวลผลต่อไป เมื่อทำการอ่านค่าของการนับเข้าไปแล้ว ตัวควบคุมก็จะส่งสัญญาณมาเคลียร์วงจรถ่ายสำหรับการเริ่มต้นอ่านข้อมูลครั้งต่อไป ดังรูปที่ 3.9



รูปที่ 3.8

แสดงการ เปลี่ยนความถี่จากขบวนการ เป็นค่าการนับในวงจรมับ

การคำนวณช่วงเวลาในการเปิดเกต ทำได้ดังนี้

1. จากขบวนการค่าอุณหภูมิสูงสุดมีค่าเท่ากับ 100°C
2. เมื่อผ่านส่วนที่แปลงค่าอุณหภูมิเป็นแรงดันไฟฟ้า ซึ่งมีค่าระดับแรงดันต่ออุณหภูมิเท่ากับ 10 มิลลิโวลต์ ต่อ องศาเซลเซียส ดังนั้นที่อุณหภูมิ 100 องศาเซลเซียส จะได้ค่าแรงดันไฟฟ้าเท่ากับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$10 \text{ mV/}^{\circ}\text{C} * 100^{\circ}\text{C} = 1 \text{ V}$$

3. จากนั้นแปลงระดับแรงดันเป็นความถี่ โดยวงจรแปลงระดับแรงดันเป็นความถี่ ซึ่งมีอัตราการเปลี่ยนแปลงความถี่ ต่อ การเปลี่ยนแปลงระดับแรงดัน เป็น 1 กิโลเฮิรต์ส ต่อ โวลต์ ดังนั้นความถี่เอาท์พุทของวงจร เมื่ออุณหภูมิมีค่า 100 องศาเซลเซียสมีค่าเท่ากับ 1 กิโลเฮิรต์ส

4. ค่าการนับของวงจรมับ จะแสดงค่าอุณหภูมิ โดยจะมีค่าของ 1 การนับต่ออุณหภูมิ 1 องศาเซลเซียส จากอุณหภูมิ 100 องศาเซลเซียส ค่าของการนับจะเท่ากับ 100 นั่นคือมีการทริกวงจรมับ 100 ครั้ง จากความถี่ 1 กิโลเฮิรต์สมีคาบเวลาเป็น 1 มิลลิวินาที ดังนั้น

$$\begin{aligned} \text{จำนวนเวลาที่ใช้ในการเปิด GATE} &= \text{จำนวนครั้งในการนับ} * \text{คาบเวลาของลูกคลื่น} \\ &= 100 * 1 \text{ ms} = 0.1 \text{ วินาที} \end{aligned}$$

ส่วนของตัวควบคุมแบบดิจิทัล

ประกอบไปด้วย ตัวไมโครคอนโทรลเลอร์ 8031 หน่วยความจำรอมสำหรับโปรแกรม ขนาด 4 กิโลไบต์ หน่วยความจำแรมสำหรับข้อมูล ขนาด 2 กิโลไบต์ ส่วนรับข้อมูลจากผู้ใช้ (Keyboard) ส่วนแสดงผลเลข 6 หลัก (Display)

1. ไมโครคอนโทรลเลอร์ 8031

ใช้ 8031 เป็นหน่วยประมวลผลกลาง โดยจะทำหน้าที่ในการคำนวณค่าต่างๆ และ การควบคุมการทำงานของระบบการควบคุม

2. หน่วยความจำ

แบ่งเป็น

หน่วยความจำรอม (ROM) เป็นหน่วยความจำสำหรับเก็บค่าโปรแกรมการทำงานของตัวไมโครคอนโทรลเลอร์ 8031 มีแอดเดรสการทำงานอยู่ที่ตำแหน่ง 0000-0FFF

หน่วยความจำแรม (RAM) เป็นหน่วยความจำสำหรับเก็บค่าข้อมูลของอุณหภูมิควบคุม มีแอดเดรสการทำงานอยู่ที่ตำแหน่ง 0000-07FF เนื้อที่ของข้อมูลที่เก็บภายในแรม จะถูกแบ่งออกเป็น ส่วน ๆ โดยแต่ละส่วนจะเก็บข้อมูลในการทำงานของแต่ละโปรแกรม แต่ละโปรแกรมจะแบ่งย่อยออกเป็นช่วงต่าง ๆ แต่ละช่วงจะประกอบไปด้วย ค่าของช่วงเวลาเป็นนาฬิกา วินาที และค่าของอุณหภูมิควบคุม เป็นหลักหน่วย หลักสิบ หลักร้อย

3. ส่วนรับข้อมูลจากผู้ใช้ (Keyboard)

มีคีย์ทั้งหมด 16 คีย์ ใช้สำหรับตั้งโปรแกรม ค่าของช่วงเวลา ค่าของอุณหภูมิ และรับคำสั่งจากผู้ใช้ ต่อพ่วงอยู่กับพอร์ท 0 ของ 8031 โดยผ่านบัฟเฟอร์ซึ่งมีตำแหน่งแอดเดรส 3000

4. ส่วนแสดงผลเลข 6 หลัก (Display)

ใช้สำหรับการแสดงผลการคำนวณจากการประมวลผล และ ติดต่อกับผู้ใช้ขณะป้อนข้อมูล แต่ละหลักของส่วนแสดงผลจะต่อพ่วงอยู่กับพอร์ท 0 ของ 8031 โดยผ่านตัวแลทซ์ข้อมูลซึ่งมีตำแหน่งแอดเดรส 3000-3005

5. ส่วนการรับข้อมูลจากขบวนการ

เป็นส่วนของบัฟเฟอร์ที่เชื่อมโยงระหว่างวงจรมับความถี่ กับ พอร์ท 0 ของ 8031 ใช้อ่านค่าอุณหภูมิจากขบวนการเพื่อนำไปใช้ในการคำนวณประมวลผล มีตำแหน่งแอดเดรส 3001

6. ส่วนส่งข้อมูลกลับไปยังขบวนการ เพื่อควบคุม

เป็นส่วนซึ่งจะนำผลลัพธ์ที่ได้จากการประมวลผลของตัวควบคุมส่งกลับไปเพื่อควบคุมอุณหภูมิของขบวนการ เป็นส่วนที่สามารถต่อเพิ่มออกทางพอร์ต 1 ของ 8๒๓1ได้ เนื่องจากไม่มีขบวนการจริง ๆ ให้ความคุม จึงทำส่วนนี้ในลักษณะการจำลองการควบคุม โดยใช้ซอฟต์แวร์ และ แสดงผลทางส่วนแสดงผล



บทที่ 4

การทดลองและผลการทดลอง

ในบทนี้ ประกอบด้วย การทดสอบผลการทำงานของโครงงานนี้ 2 ส่วนใหญ่ ๆ คือ ส่วนการวัดคุณสมบัติและแปลงเป็นสัญญาณดิจิทัล กับ ส่วนของตัวควบคุม

4.1/ ส่วนการวัดคุณสมบัติและแปลงเป็นสัญญาณดิจิทัล

แบ่งการทดลองออกเป็นขั้นตอนต่าง ๆ ดังนี้
ขั้นที่ 1 ทดสอบและปรับค่าวงจรแปลงระดับแรงดันเป็นความถี่ ให้ได้อัตราการแปลงความถี่ต่อการเปลี่ยนแปลงของระดับแรงดัน เท่ากับ 1 กิโลเฮิรตซ์ ต่อ โวลต์ จากรูปที่ 3.8 ความสัมพันธ์ระหว่างความถี่เอาต์พุต กับ ระดับแรงดันอินพุต ของวงจร เป็นไปตามสมการ

$$f_{OUT} = \frac{-V_{IN} * R_S * 1}{2.09 * R_{IN} * R_T * C_T}$$

เราใช้ค่า $R_T = 6.8 \text{ K}$ $C_T = 0.01 \text{ uF}$

$R_S = 12.1 \text{ K (คงที่) + 5 K (ปรับค่าได้)}$

$R_{IN} = 100 \text{ K}$

จากนั้น ทำการปรับค่าความต้านทาน R_S และปรับค่าออฟเซ็ทในวงจรอินทิเกรเตอร์ ให้ได้ อัตราการเปลี่ยนแปลงความถี่ต่อการเปลี่ยนแปลงของระดับแรงดันเท่า

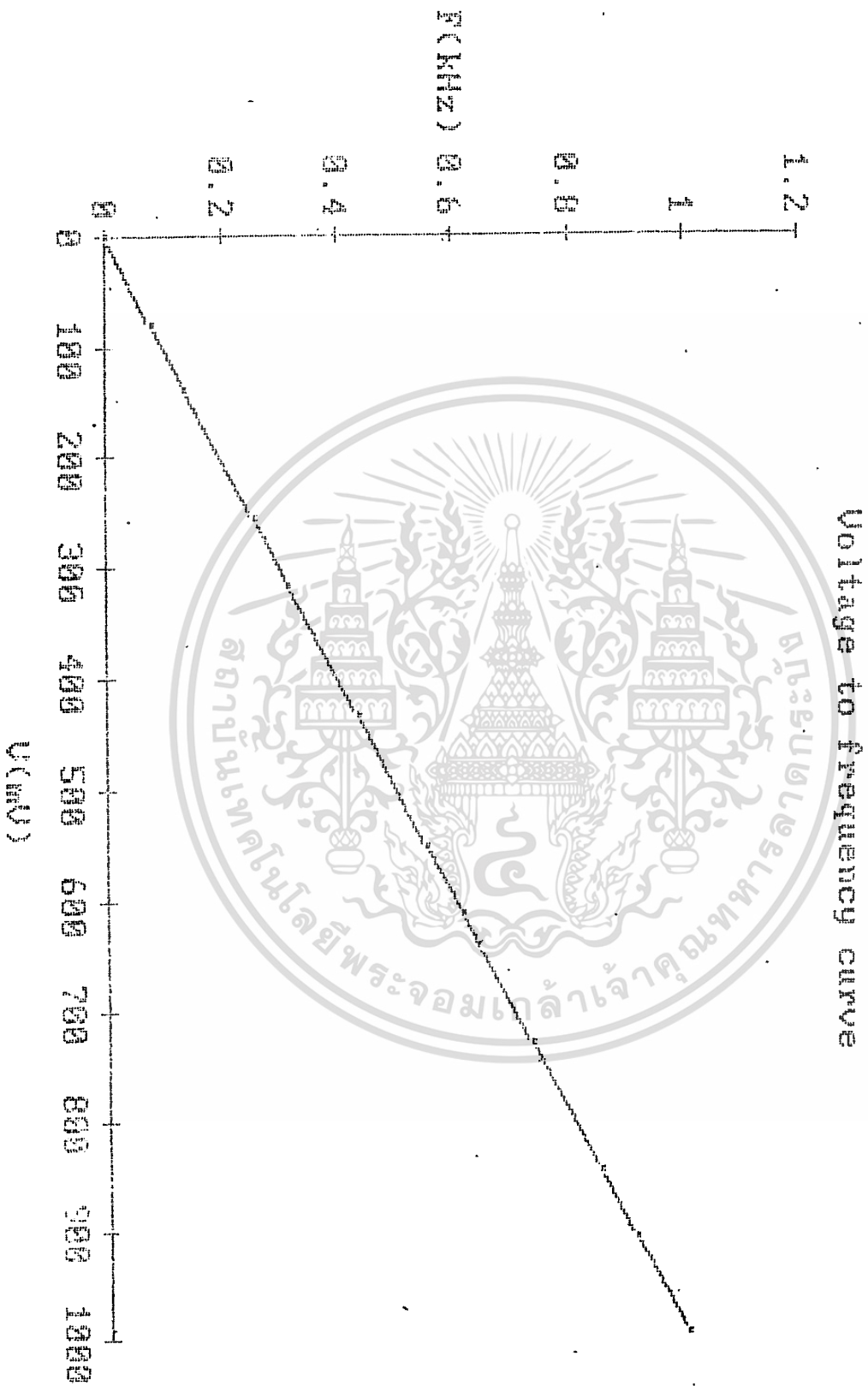
กับ 1 กิโลเฮิรตซ์ ต่อ โวลท์ ซึ่งได้ผลการทดลองดังตารางที่ 4.1 | และนำตารางมาเขียนกราฟได้ดังรูปที่ 4.2 |

ระดับแรงดัน (mV)	ความถี่ (KHz)	ความผิดพลาด (%)
0	0	0
80	.078	-2.5
138	.138	0
256	.258	0.78
316	.318	0.63
433	.438	1.15
552	.556	0.72
612	.619	1.14
732	.741	1.23
849	.861	1.41
909	.921	1.32
999	1.011	1.20

รูปที่ 4.1 |

ตารางผลการทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นที่ 2 ในขั้นนี้จะทำการปรับค่าวงจรชดเชยอุณหภูมิ ให้ได้อัตราการเปลี่ยนแปลงของระดับแรงดันต่ออุณหภูมิเป็น $10 \text{ mV}/^{\circ}\text{C}$

การทดลองทำโดย

ก) นำเทอร์โมคัปเปิลออก แล้วต่อกราวด์เป็นอินพุทของวงจรถาย เพื่อให้แรงดันของวงจรถายของเทอร์โมคัปเปิลมีค่าเป็นศูนย์ (อาจมีค่าไม่เป็นศูนย์ แต่ใกล้ ๆ ศูนย์ เนื่องจากออฟเซ็ทของออปแอมป์ ค่าออฟเซ็ทนี้ สามารถชดเชยได้จากการปรับค่าศูนย์)

ข) ใช้วงจรถดเชยอุณหภูมิ ทำการวัดค่าอุณหภูมิ จากการต้มน้ำ แล้วจัดค่าความถี่เอาท์พุทของวงจรแปลงแรงดันเป็นความถี่ (ความถี่ 0.01 กิโลเฮิรตซ์ แทนอุณหภูมิ 1°C) เทียบกับอุณหภูมิที่วัดได้จากเทอร์โมมิเตอร์

ค) ทำการปรับแต่ง (calibration) ค่า VR_1 และ VR_2 เพื่อให้อุณหภูมิที่วัดได้จากวงจรถดเชยอุณหภูมินี้ มีค่าใกล้เคียงกับเทอร์โมมิเตอร์มากที่สุด การปรับ VR_1 นี้ คือการปรับสเปน (span) และการปรับ VR_2 คือการปรับค่าศูนย์ (zero) ของวงจร

จากการปรับแต่งวงจรถดเชยอุณหภูมิ ได้ผลการทดลองออกมา ดังตารางรูปที่ 4.3।

อุณหภูมิ ($^{\circ}\text{C}$)	ความถี่ (KHz)	ความผิดพลาด (%)
70	.70	0.0
69	.69	0.0
68	.68	0.0
67	.67	0.0
66	.65	-1.5
65	.64	-1.5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

64	.63	-1.5
63	.62	-1.6
62	.62	0.0
61	.61	0.0
60	.60	0.0
59	.59	0.0
58	.58	0.0
57	.57	0.0
56	.55	-1.8
55	.54	-1.8
54	.52	-3.7
53	.50	-5.6
52	.49	-5.7
51	.48	-5.8
50	.47	-6.0

รูปที่ 4.3

ตารางผลการทดลองของวงจรถดเชยขุดหลุม

ขั้นที่ 3 หลังจากผ่าน 2 ขั้นตอนแรกมาแล้ว ในขั้นตอนนี้ จะนำเอาเทอร์โมคิปเปิลต่อเข้าไปเป็นอินพุตของวงจรถายสัญญาณ แล้วนำเทอร์โมคิปเปิลวัดอุณหภูมิจากการต้มน้ำ ส่วนวงจรถดเชยขุดหลุมที่ปรับแต่งแล้วจะวัดอุณหภูมิห้อง จากนั้น ทำการปรับแต่งค่าการขยายของวงจรถาย และ ค่า VR_2 แล้วเปรียบเทียบความถี่จากวงจรแปลง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แรงดันเป็นความถี่ กับอุณหภูมิที่วัดจากเทอร์โมมิเตอร์ ปรับให้ได้ค่าใกล้เคียงกับเทอร์โมมิเตอร์มากที่สุด

ผลการวัดอุณหภูมิจากเทอร์โมคัปเปิล ซึ่งมีการชดเชยอุณหภูมิแล้ว แสดงดังตารางรูปที่ 4.4

อุณหภูมิ (°C)	ความถี่ (KHz)	ความผิดพลาด (%)
70	.70	0.0
69	.70	-1.4
68	.69	-1.5
67	.67	0.0
66	.66	0.0
65	.65	0.0
64	.65	-1.6
63	.63	0.0
62	.61	-1.6
61	.60	-1.6
60	.59	-1.7
59	.58	-1.7
58	.55	-5.2
57	.54	-5.3
56	.53	-5.4
55	.52	-5.5
54	.50	-7.4
53	.49	-7.5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

52	.48	-7.7
51	.47	-7.8
50	.46	-8.0

รูปที่ 4.4

ตารางแสดงการวัดอุณหภูมิของทั้งหมด

4.2 ส่วนของตัวควบคุมแบบคิจิตอล

ในส่วนนี้ นับว่าเป็นส่วนที่ทำการทดลองยากที่สุด เนื่องจากไม่มีอุปกรณ์ที่จะลอกเลียนแบบการทำงานของ 8031 ได้ ดังนั้น เราจึงต้องทำการแปลงค่ารหัสโปรแกรมให้เป็นรหัสออปโคด (op code) ทั้งหมด จากนั้น นำรหัสออปโคดที่ได้ บ้อนลงในแรมแพค (ram pack) โดยใช้บอร์ด เอ็มพีเอฟ-1 (MPF-1 single board) เป็นตัวบ้อน แล้วจึงนำแรมแพคใส่ลงบนบอร์ด 8031 เพื่อทำการเอ็กซีคิวท์ (execute) โปรแกรมจากขั้นตอนดังกล่าว จะเห็นว่า เราจะต้องเสียเวลาในการทำเป็นอย่างมาก เนื่องจากถ้า เอ็กซีคิวท์โปรแกรมแล้ว ผลที่ได้ไม่เป็นไปตามที่ต้องการ เราก็จะต้องทำการแก้ไขหาข้อผิดพลาด (debug program) ซึ่งเราจะต้องกระทำด้วยตัวเอง หลังจากนั้น ก็ต้องเริ่มต้นแปลงรหัสให้เป็นรหัสออปโคดใหม่อีกครั้งหนึ่ง เป็นเช่นนี้เรื่อยไป จนกระทั่ง การเอ็กซีคิวท์โปรแกรม ให้ผลตามที่เราต้องการ ซึ่งขั้นตอนเหล่านี้ กระทำได้ลำบาก และไม่สะดวกเป็นอย่างมาก โดยเฉพาะอย่างยิ่ง การบ้อนรหัสออปโคดเป็นจำนวนมากลงในแรมแพค ซึ่งต้องอาศัยความละเอียดรอบคอบ ระมัดระวัง ตลอดจนความอดทนเป็นอย่างมาก

สำหรับผลการเอ็กซีคิวท์โปรแกรมในโครงการนี้ ผลปรากฏว่า ในส่วนรับ

ข้อมูลจากคีย์บอร์ด และส่วนแสดงผล สามารถทำงานได้ 100 เปอร์เซ็นต์ แต่สำหรับ
ในส่วนการควบคุมขบวนการ ไม่สามารถทำงานได้ เนื่องจาก การตรวจสอบหาที่ผิด
กระทำได้ลำบากและยุ่งยากนั่นเอง



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

บทวิจารณ์ และ บทสรุป

5.1/บทวิจารณ์

ในการเลือกใช้ไมโครคอนโทรลเลอร์ 8031 เป็นส่วนประมวลผล มีข้อดี และข้อเสียดังต่อไปนี้

5.1.1 ข้อดี

- การกระทำทางคณิตศาสตร์ในคำสั่งคูณและหารของ 8031 กระทำทางฮาร์ดแวร์ โดยใช้คำสั่งเดียว ซึ่งง่ายและสะดวกรวดเร็ว

- มีอินพุตและเอาต์พุตพอร์ทถึง 4 พอร์ท พอร์ทอนุกรมที่สามารถรับและส่งข้อมูลแบบอนุกรม และไทม์เมอร์เคาท์เตอร์ภายในขนาด 16 บิตจำนวน 2 ตัว อยู่ในตัว 8031 เองเลย โดยไม่ต้องเพิ่มอุปกรณ์สนับสนุน

5.1.2 ข้อเสีย

การตรวจสอบหาข้อผิดพลาดจากการเอ็กซ์ซิวิตว์โปรแกรม ทำได้ยุ่งยาก และลำบากเนื่องจากไม่มี ซอฟต์แวร์ (software) ที่ลอกเลียนแบบการทำงานของ 8031 ได้ร้อยเปอร์เซ็นต์

นอกจากนี้ ยังมีปัญหาต่าง ๆ อื่น ๆ อีกที่พบในการทำโครงการนี้ อันได้แก่ การคำนวณค่าอุปกรณ์ต่าง ๆ เช่น ค่าความต้านทาน ค่าตัวเก็บประจุ ในวงจรต่าง ๆ ค่าที่คำนวณได้ ใช้ได้แค่เป็นแนวทางเท่านั้น เพราะ เมื่อต่อวงจรโดยใช้ค่าที่คำนวณได้ เราต้องมีการปรับค่าที่คำนวณได้อีกเล็กน้อย เพื่อให้ผลที่ได้ออกมามีค่าถูกต้องตรงตามทฤษฎี นอกจากนั้น ยังมีปัญหาเรื่อง สัญญาณรบกวน (noise) จากภายนอก ซึ่งก็สามารถแก้ไขได้โดยใช้วงจร โลว์พาสฟิลเตอร์ (low pass filter) ซึ่งบางทีก็ไม่สามารถกรองสัญญาณรบกวนออกได้ทั้งหมด

5.2 | บทสรุป

เราสามารถใช้ประโยชน์จากตัวไมโครคอนโทรลเลอร์ 8031 ในงานควบคุมได้อย่างมีประสิทธิภาพ

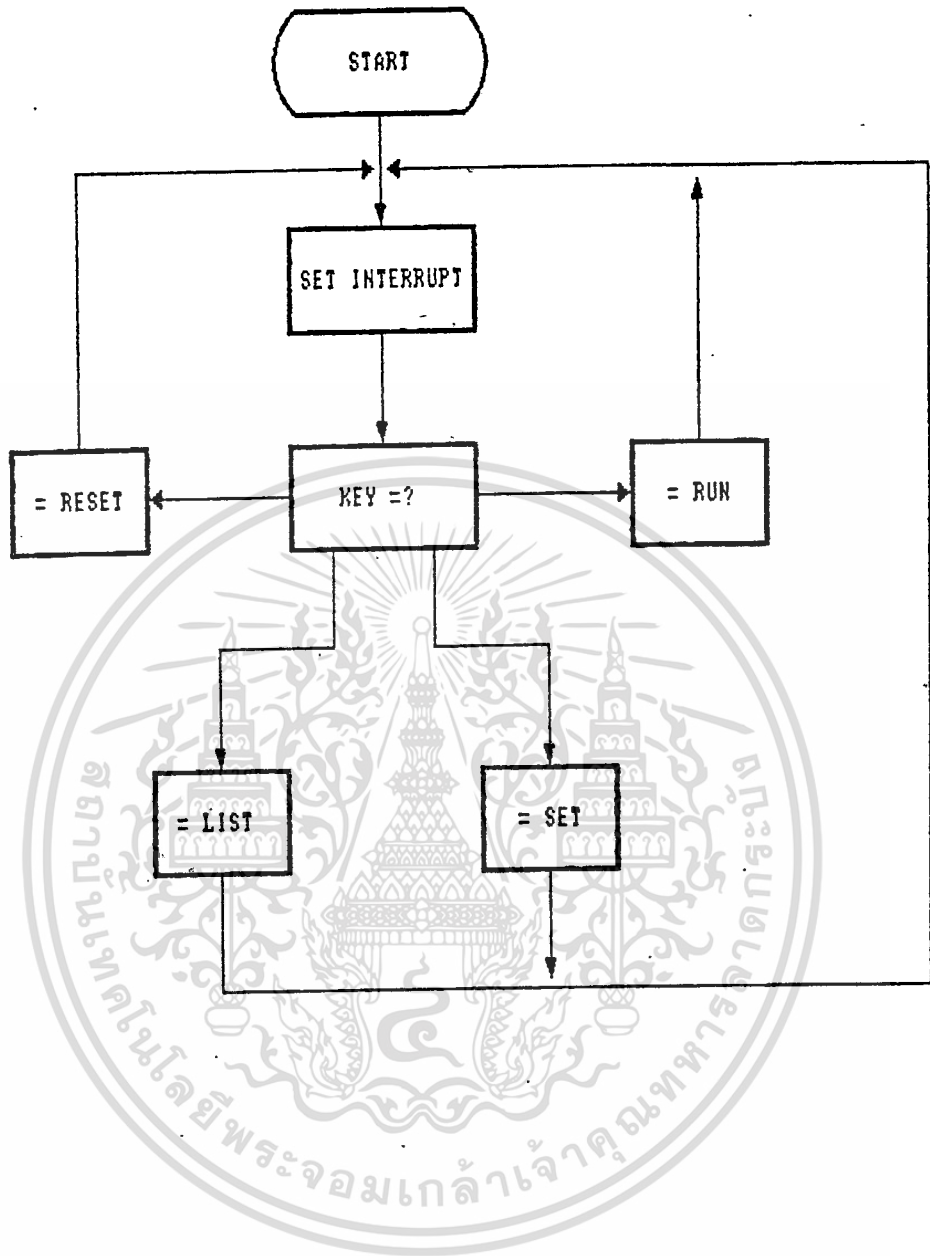
โครงการนี้ สามารถพัฒนาต่อไปได้อีก โดยการปรับปรุงโปรแกรมในส่วนควบคุม (โหมด run) และเพิ่มเติมในส่วนของการรับ-ส่งข้อมูลแบบอนุกรม เพื่อแสดงกราฟผลการทำงานของขบวนการบนจอภาพ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

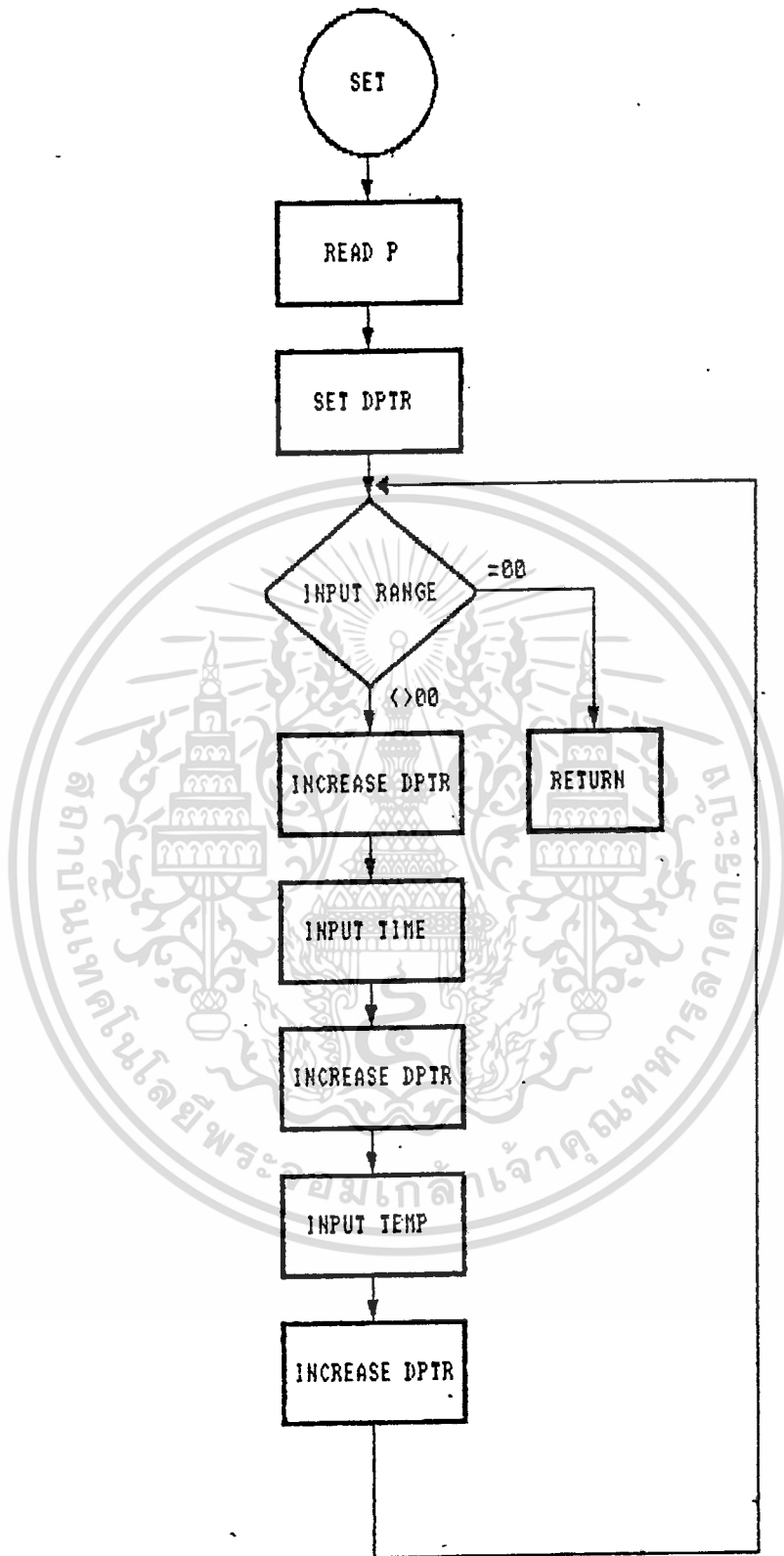


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ก-1
แสดงโฟลว์ชาร์ทของโปรแกรมหลัก (MAIN PROGRAM)

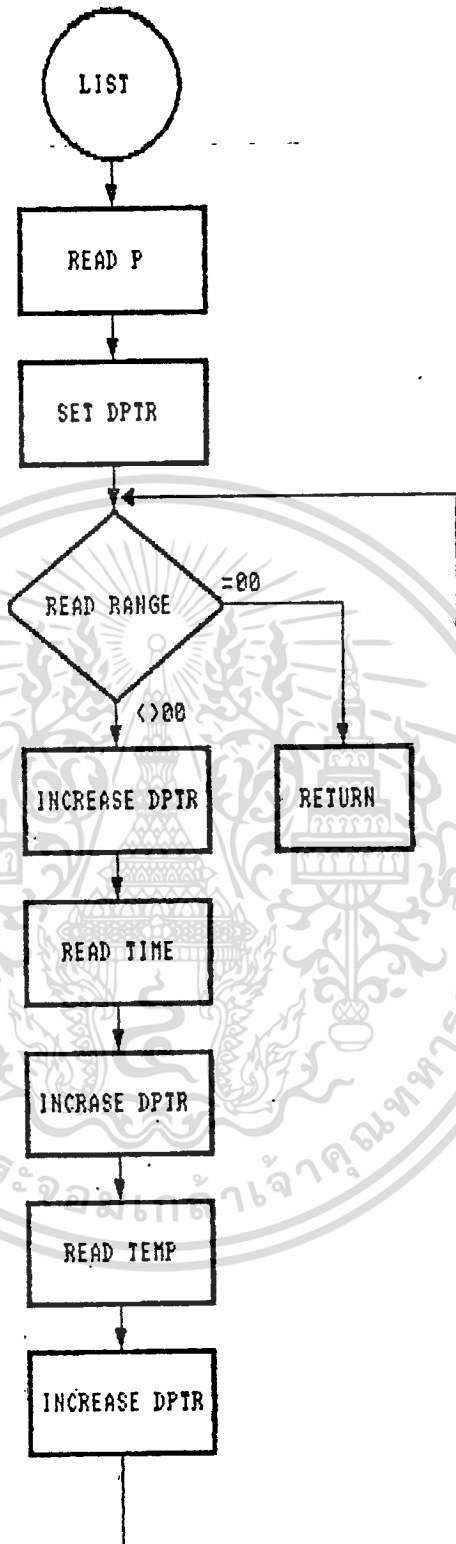
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ก-2

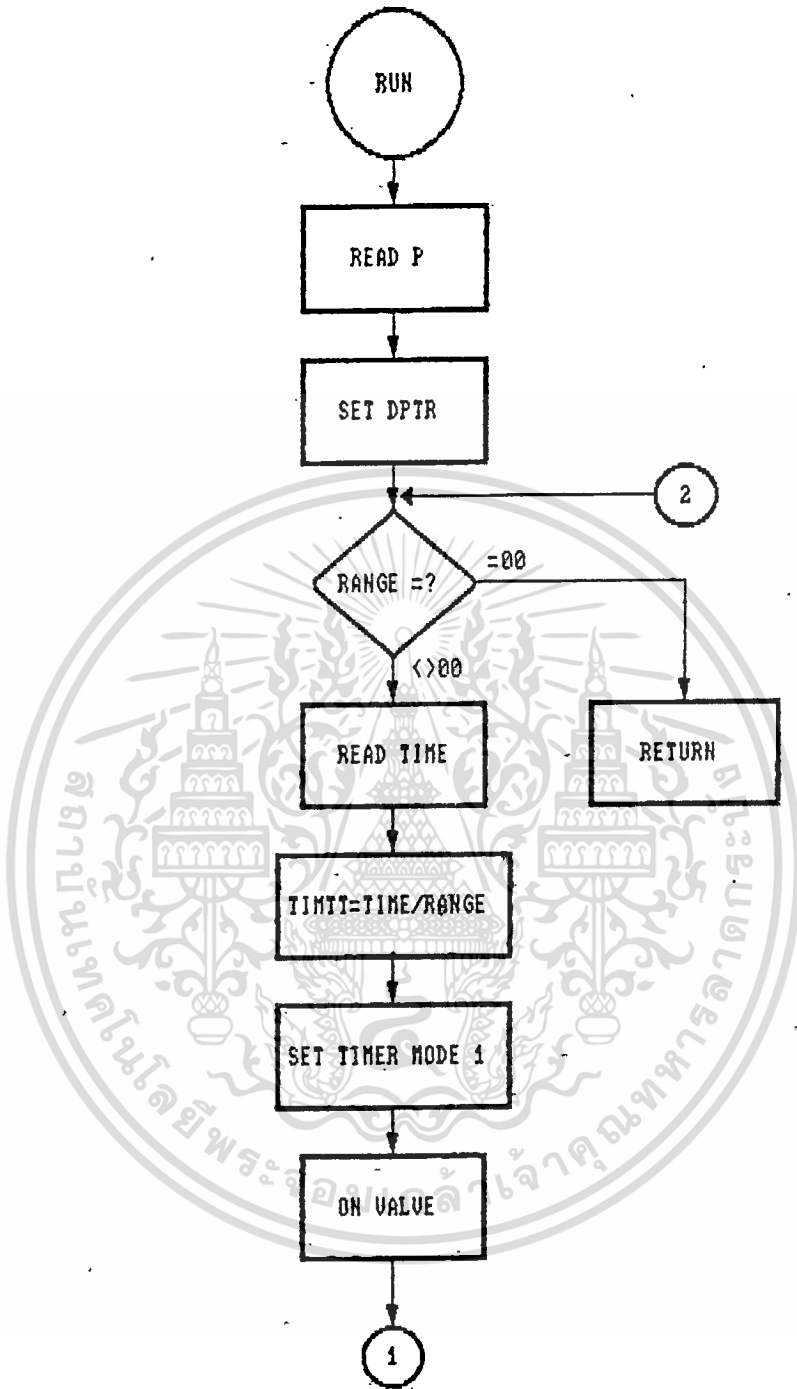
แสดงโปรแกรมในโหมด เซ็ท (SET)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



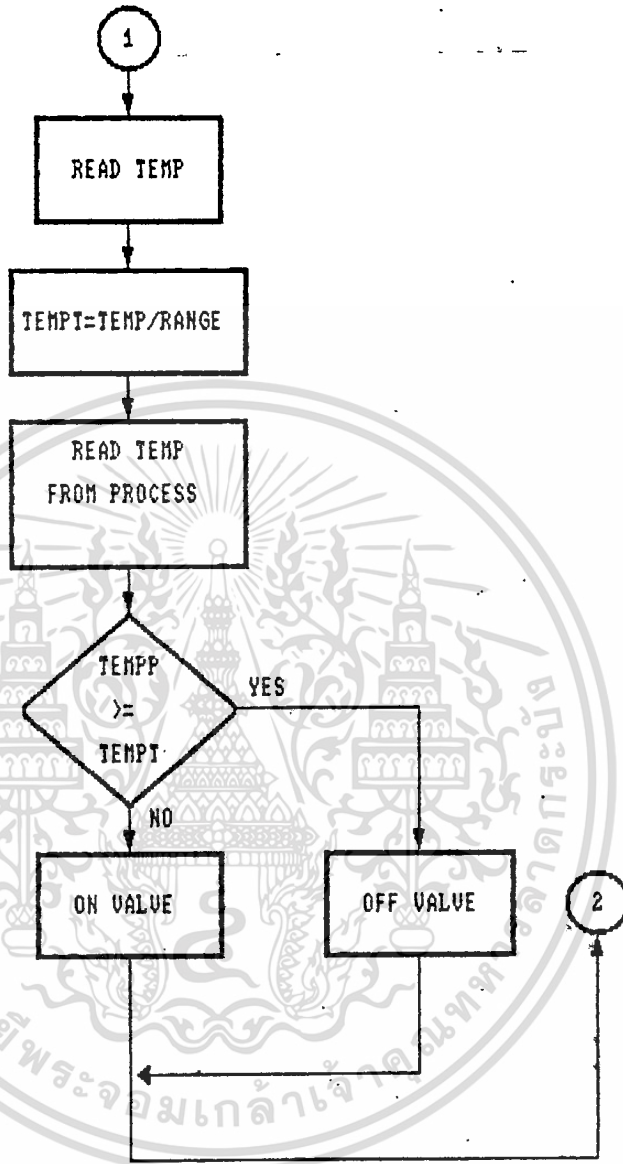
รูปที่ ก-3
แสดงโปรแกรมในโหมด ลิสต์ (LIST)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

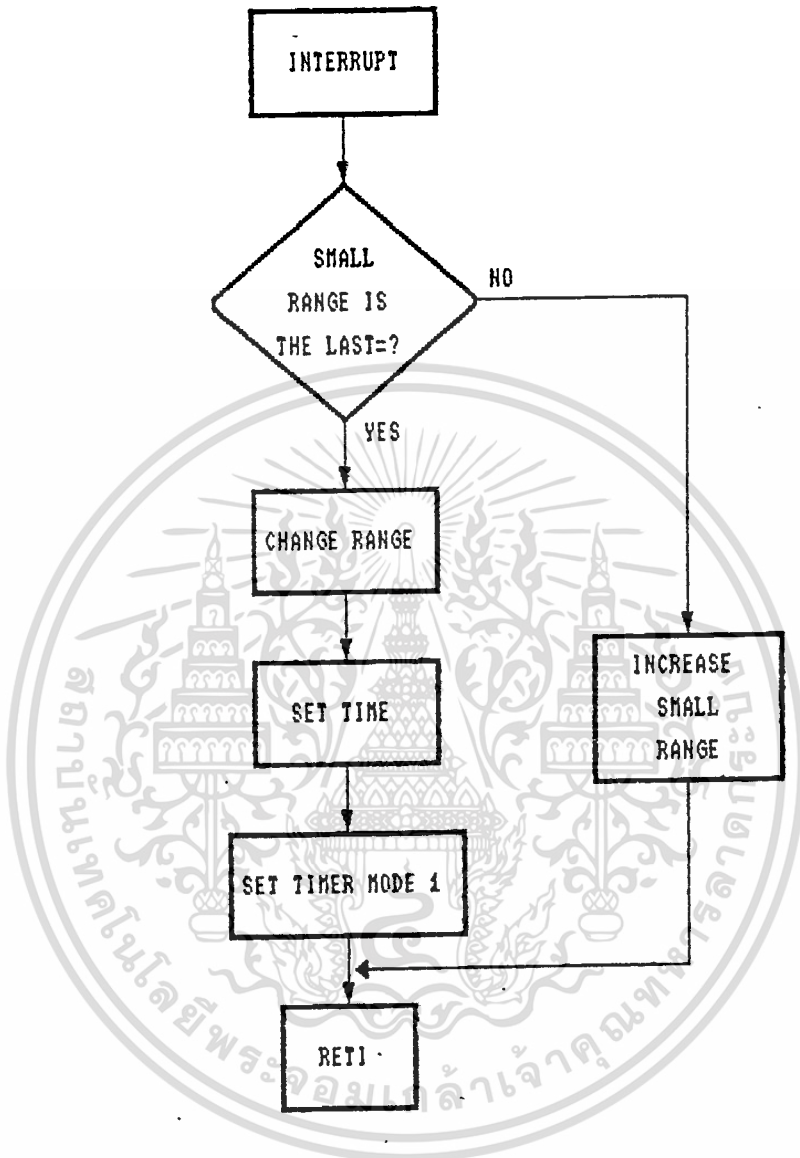


รูปที่ ก-4
แสดงโปรแกรมรันในโหมด รัน (RUN)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



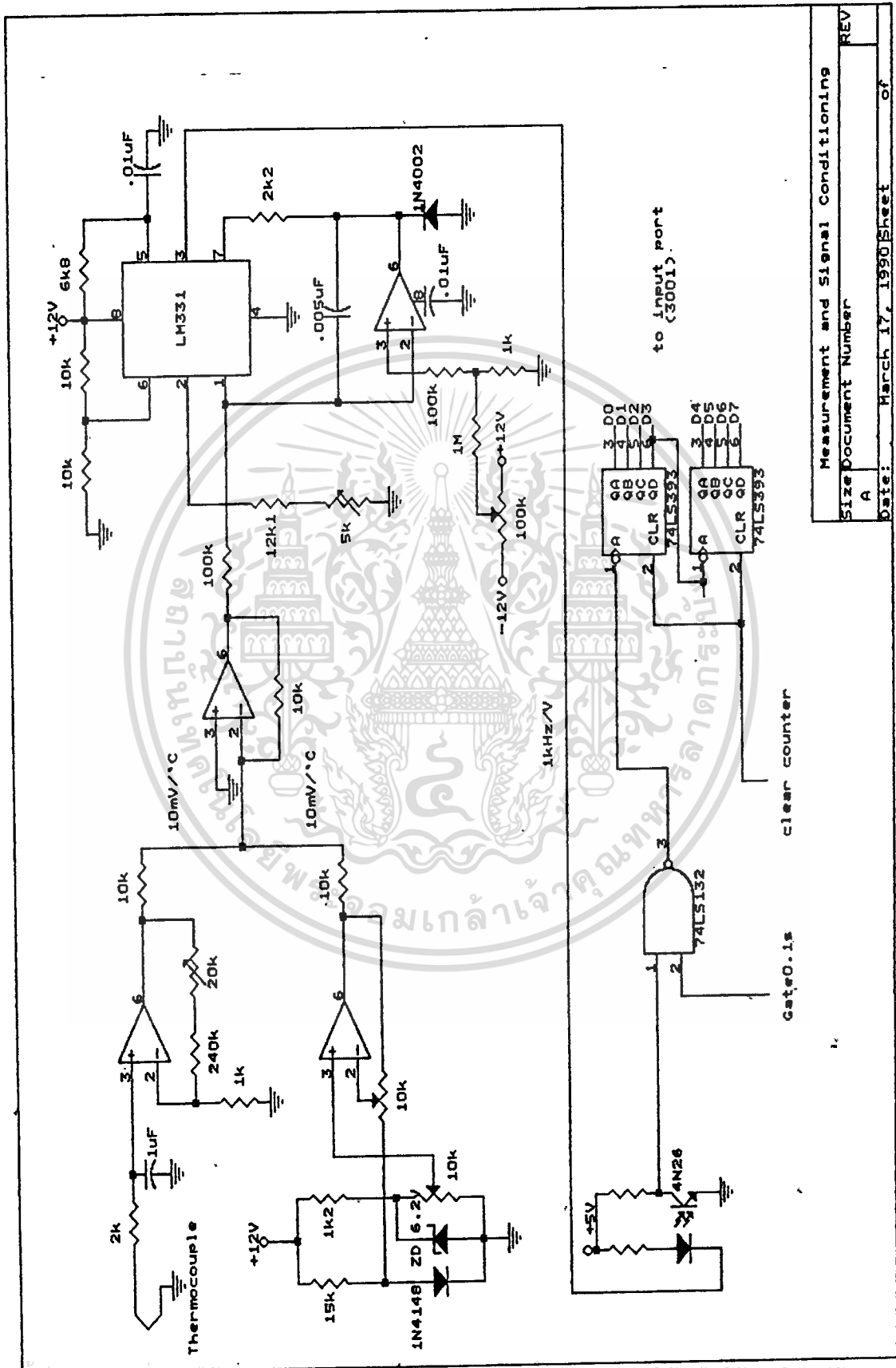
รูปที่ ก-5

แสดงไฟล์ชาร์ทในสับรุตินอินเทอร์รัพท์ (INTERRUPT SUBROUTINE)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

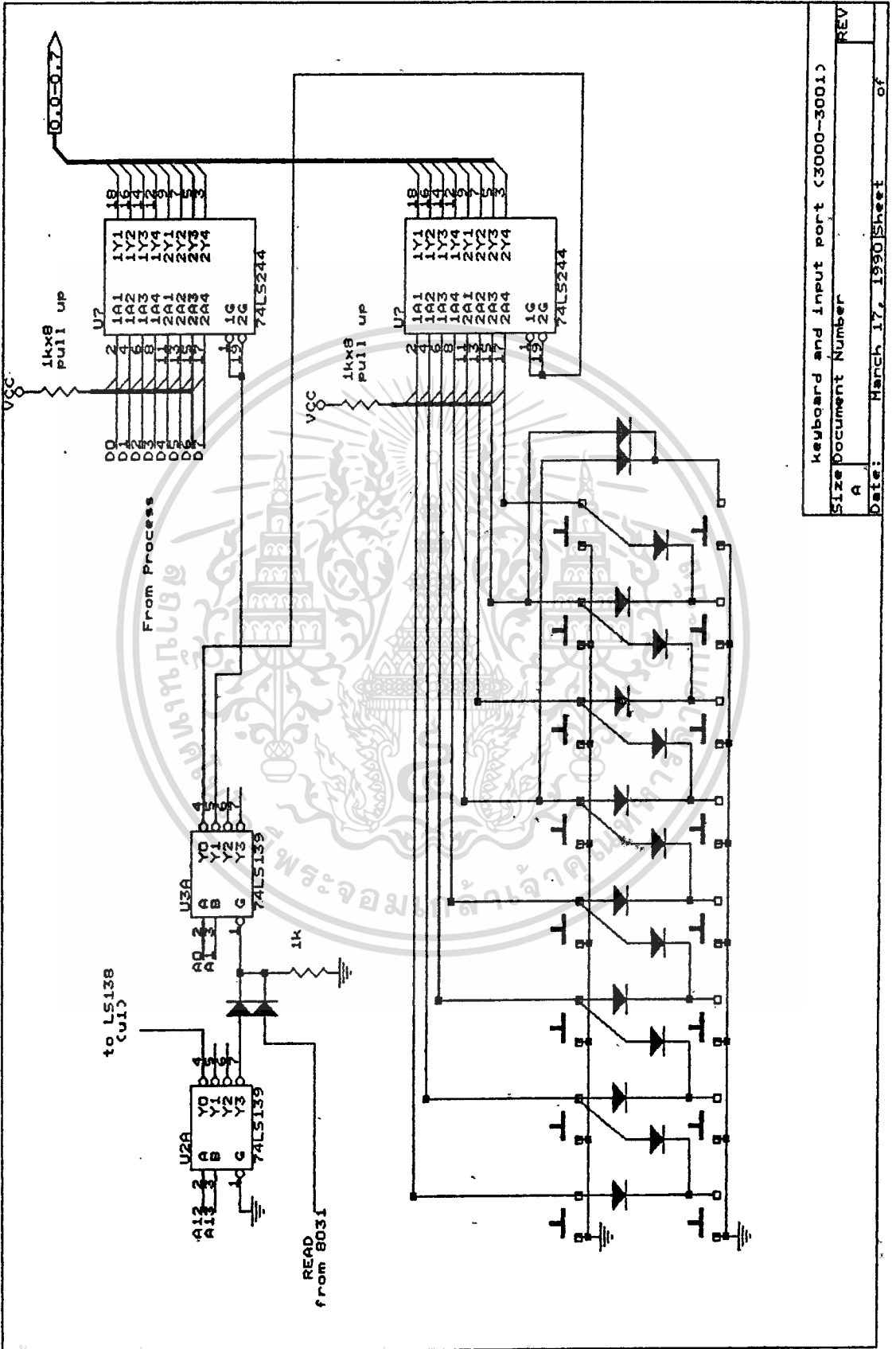


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Measurement and Signal Conditioning	
Size Document Number	REV
A	
Date:	March 17, 1990 Sheet of

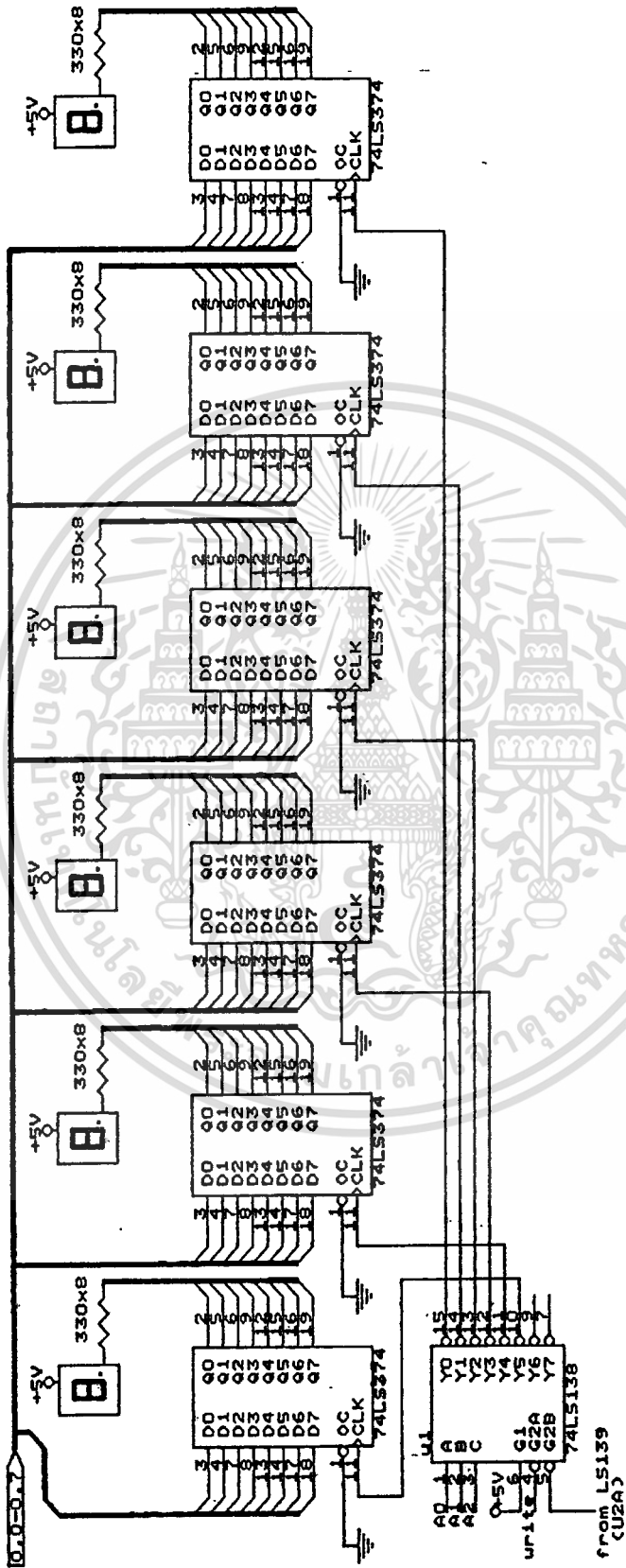
เอกสารนี้เป็นเอกสารทสมวนเวลาหรับการใชงานเพื่อการศึกษาเท่านั้น ไมอนุญาตใหนำไปใชประโยชน์ดานการค้า
ไมวากรณีใดๆทั้งสิ้น อีกทั้งยังมีใหัดดแปลงเนื้อหา และตองอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช



Keyboard and input port (3000-3001)	
Size	Document Number
A	REV
Date:	March 17, 1990
Sheet	of

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

data bus



Display (3000-3005)
Size Document Number
A
Date: March 17, 1990 Sheet of
REV

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิติกรรมประกาศ

ผู้จัดทำขอขอบพระคุณ ท่านอาจารย์ วันชัย ธีรรุจา ที่คอยช่วยเหลือ ให้คำปรึกษาในทุก ๆ ด้าน และคอยกระตุ้นเตือนผู้จัดทำให้มีความกระตือรือร้นอยู่เสมอ ขอขอบพระคุณ อาจารย์ เกียรติวรรณ ทรงลัศย์ ที่ให้ความสะดวกในการเบิกอุปกรณ์ในการทำโครงการ ขอขอบคุณเพื่อน ๆ ที่คอยเป็นกำลังใจ และให้ความสนุกสนาน ตลอดเวลาการทำงาน ขอขอบคุณน้องตารารัตน์ ที่ช่วยในการจัดทำปฏิทินฉบับนี้ สุดท้ายที่จะลืมไม่ได้คือ บพภกริชของผู้จัดทำ ที่ท่านได้ให้กำเนิด อบรมล้างสอน เลี้ยงดูผู้จัดทำมาตลอด



หนังสืออ้างอิง

1. National Semiconductor Corporation, "Linear Databook", 1982.
2. National Semiconductor Corporation, "Logic Databook" Volume 11, 1984.
3. Handbook MCS-51, Intel Corporation, 1985.
4. Curtis D. Johnson, "Microprocessor Based Process Control", Prentice-Hall, Inc., 1984.
5. จักร์กฤษ ชัยชนะสุวัฒน์, "เครื่องวัดอุณหภูมิ", วารสารเซมิคอนดักเตอร์อิเล็กทรอนิกส์, ฉบับที่ 74, 2530, หน้า 146-147.