การจำลองกระบวนการทางความร้อน โดย ไมโครคอนโทรลเลอร์

THERMAL PROCESS SIMULATOR BY MCS - 51

โดย

นายถวัลย์ วรจินต์

นายทวีเกียรติ สดสิริสวัสดิ์

นายพิษณุรัตน์ เมนาคม

วัน เดือน ปี 1 ธ.ค. 2540

เลขทะเบียน 037158

เลขเรียกหนังสือ T 38251 ถ247 ถ367 ก

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตร์บัณฑิต

สาขาวิศวกรรมการวัดคุมทางอุตสาหกรรม

สถาบันเทคโนโลยีพระจอมเกล้า เจ้าคุณทหาร ลาดกระบัง

ปีการศึกษา ๒๕๓๘

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

King Mongkut's Institute of Technology
LADKRABANG
Instrumentation Department

THERMAL PROCESS SIMULATOR BY MCS - 51

โดย

นายถวัลย์ วรจินต์ ๓๖๐๑๓๒๘๕
นายทวีเกียรติ สดสิริสวัสดิ์ ๓๖๐๑๓๒๘๖
นายพิษณุรัตน์ เมนาคม ๓๖๐๑๓๒๙๕

อาจารย์ที่ปรึกษา
(.....)
อาจารย์ วิริยะ กองรัตน์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การจำลองกระบวนการทางความร้อน โดย ไมโครคอนโทรลเลอร์
THERMAL PROCESS SIMULATOR BY MCS - 51

โดย นายถวัลย์ วรจินต์ รหัส ๓๖๐๑๓๒๘๕

นายทวีเกียรติ สดสิริสวัสดิ์ รหัส ๓๖๐๑๓๒๘๖

นายพิษณุรัตน์ เมนาคม รหัส ๓๖๐๑๓๒๙๕

อาจารย์ที่ปรึกษา อาจารย์ วิริยะ กองรัตน์

บทคัดย่อ

ปัจจุบันเทคโนโลยีการควบคุมกระบวนการผลิต ได้พัฒนาขึ้นมากกว่าสมัยก่อนมาก ส่งผลให้ไม่มีการปรับตัวของผู้ปฏิบัติการ ในการที่จะใช้งานตัวควบคุมต่าง ๆ ให้ได้เต็มความสามารถของอุปกรณ์นั้น ๆ ได้ 100%

โครงงาน **"การจำลองกระบวนการทางความร้อน"** เป็นโครงงาน ที่ได้นำเอาทฤษฎีคุณสมบัติทางกายภาพของกระบวนการทางพลศาสตร์ มาจำลองทางคณิตศาสตร์และสังเคราะห์ให้อยู่ในรูปของซอฟแวร์ภาษาแอสเซมบลี ซึ่งใช้ร่วมกับไมโครคอนโทรลเลอร์ MCS - 51 เพื่อวิเคราะห์คุณสมบัติของกระบวนการที่ใช้ร่วมกับระบบควบคุม และผลการควบคุมล่วงหน้า ก่อนการก่อสร้างกระบวนการและติดตั้งอุปกรณ์ควบคุมในกระบวนการ

ตัวจำลองยังสามารถใช้งานร่วมกับ ตัวควบคุมจากภายนอก และมีการแสดงผลทางคอมพิวเตอร์ เพื่อให้ผู้ใช้ได้สามารถศึกษาถึงธรรมชาติของกระบวนการในแต่ละชนิด ซึ่งเป็นการพัฒนาความรู้ความสามารถของผู้ใช้อีกอย่างหนึ่ง

ABSTRACT

Recently, Process control technology has developed from the last decade. Simultaneorsly the improvement of users has been cut down, the result is that workability of most of controllers not reach to 100%

"Thermal Process simulator By MCS-51" is the project by applyingthe Physical properties and Dynamics system in program computer which can done in MCS-51 for must consider how citical the performance of the control system is for the safe and profitable operation of the process.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

การที่ปริญญานิพนธ์นี้สำเร็จขึ้นมาได้ก็ด้วยความร่วมมือจากหลายฝ่ายด้วยกัน คณะผู้จัดทำต้องขอขอบคุณอย่างสูงในโอกาสนี้ด้วย โดยเฉพาะบุคคลดังต่อไปนี้

อาจารย์วิริยะ กองรัตน์ อาจารย์ที่ปรึกษาตั้งแต่การเริ่มจัดทำทั้งยังให้ความสะดวกและจัดหาอุปกรณ์ในการจัดทำด้วย

บริษัท ศิลารีเสิรช์ และบริษัท อีทีที ให้คำปรึกษาด้าน Micro Controller Board

นาย ปรีชา เนียมพลอย ผู้ช่วยหัวหน้าหมวดเครื่องมือระบบควบคุม การไฟฟ้าฝ่ายผลิตแห่งประเทศไทย โรงจักรพระนครใต้ ที่เอื้อเฟื้ออุปกรณ์เครื่องมือ ตลอดจนเป็นกำลังใจให้ทำงานสำเร็จ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำนำ

การออกแบบระบบควบคุม จำเป็นต้องทราบธรรมชาติ และคุณสมบัติต่าง ๆ ของกระบวนการอุตสาหกรรม เพื่อใช้ศึกษา และวิเคราะห์สภาพการเปลี่ยนแปลงของกระบวนการเมื่อถูกรบกวนจากการเปลี่ยนแปลงของสภาพแวดล้อมภายนอก หรือได้รับการควบคุมการศึกษาธรรมชาติและวิเคราะห์คุณสมบัติต่าง ๆ ของกระบวนการอุตสาหกรรมมี 2 วิธี คือ

1. การวิเคราะห์กระบวนการจากผลการทดลอง การวิเคราะห์กระบวนการสามารถทำได้โดยมีเครื่องมือวัด และควบคุมภายในกระบวนการอุตสาหกรรมทั้งหมด สำหรับใช้ในการทดลองโดยทดลองป้อนสัญญานเข้าและสังเกตผลการเปลี่ยนแปลงของสัญญานออกของอุปกรณ์แต่ละตัวซึ่งการวิเคราะห์กระบวนการโดยวิธีนี้ทำให้เสียเวลาและค่าใช้จ่ายมาก

2. การวิเคราะห์กระบวนโดยการคำนวณ เนื่องจาก การออกแบบระบบควบคุมส่วนใหญ่ต้องสำเร็จเรียบร้อยก่อนการก่อสร้าง และติดตั้งอุปกรณ์ควบคุมต่าง ๆ ในกระบวนการวิเคราะห์ กระบวนการโดยการทดลองป้อนสัญญานเข้าให้อุปกรณ์ทุกตัว เพื่อวิเคราะห์ผลของสัญญานออกจึงไม่สามารถทำได้ การวิเคราะห์กระบวนการอุตสาหกรรมส่วนใหญ่จึงต้องทำโดยแสดงคุณสมบัติของอุปกรณ์ภายในกระบวนการ โดยใช้สมการทางคณิตศาสตร์ต่าง ๆ เช่น สมการพีชคณิต สมการอนุพันธ์ (differential equation) และสมการแตกต่าง (difference equation) และโครงงานนี้ เป็นอีกแนวความคิดหนึ่ง ซึ่งได้จำลองคุณสมบัติทางกายภาพของกระบวนการโดยใช้ความสัมพันธ์ทางสมการทางคณิตศาสตร์ แล้วเปลี่ยนรูปให้อยู่ในไมโครคอนโทรลเลอร์ (MCS-51) เพื่อวิเคราะห์คุณสมบัติของกระบวนการทางอุตสาหกรรมโดยการพิจารณาหาคำตอบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

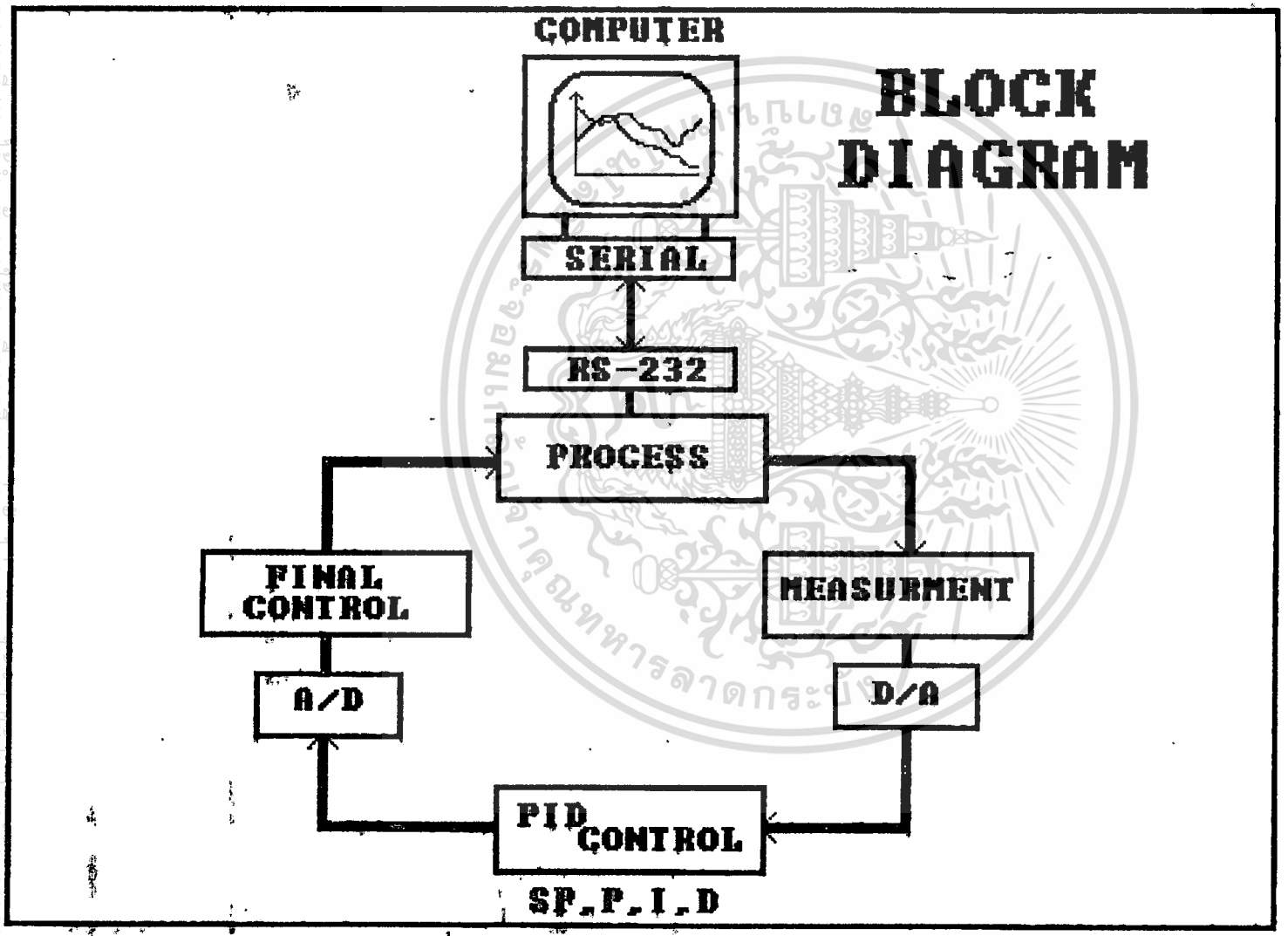
สารบัญ

	หน้า
บทคัดย่อ	
กิตติกรรมประกาศ	
บทนำ	
บทที่ 1 :ทฤษฎีที่ใช้ในการจำลองกระบวนการทางความร้อน	1 -- 24
บทที่ 2 :แนวความคิดในการสร้างโปรแกรม เพื่อใช้ในการจำลองกระบวนการทางความร้อน	25 -- 37
บทที่ 3 :การออกแบบตัวจำลองกระบวนการทางความร้อน โดยการใช้ ไมโครคอนโทรลเลอร์ (MCS -51)	40 -- 67
บทที่ 4 : การออกแบบซอฟแวร์ เพื่อใช้ในการแสดงผล ของตัวจำลองกระบวนการทางความร้อน	68 -- 74
บทที่ 5 : การทดลองและผลการทดลอง	75 -- 79
ภาคผนวก	
เอกสารอ้างอิง	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
ทฤษฎีในการสร้างตัวจำลองกระบวนการทางความร้อน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


COMPUTER
BLOCK
DIAGRAM
SERIAL
RS-232
PROCESS
FINAL
CONTROL
MEASURMENT
A/D
D/A
PID
CONTROL
SP_P_I_D

บทที่ 1

ทฤษฎีที่ใช้ในการจำลองกระบวนการทางความร้อน

การสร้างแบบจำลองทางคณิตศาสตร์

การสร้างแบบจำลองทางคณิตศาสตร์ที่น่าเชื่อถือ มีประสิทธิภาพ และสามารถทำนายการเปลี่ยนแปลงของกระบวนการได้ถูกต้อง เป็นเรื่องที่ยากที่สุด ในการวิเคราะห์และออกแบบระบบควบคุมในกระบวนการอุตสาหกรรมก่อนสร้างแบบจำลอง ทางคณิตศาสตร์ควรศึกษาและทำความเข้าใจเกี่ยวกับคุณสมบัติของกระบวนการ ดังนี้

1. เป้าหมายหรือวัตถุประสงค์การควบคุม
2. การรบกวนภายนอกที่คาดว่ากระบวนการจะได้รับและผลกระทบที่จะเกิดขึ้น
3. คุณสมบัติทางฟิสิกส์และเคมีที่เด่นและสำคัญของกระบวนการ เมื่อเข้าใจลักษณะของกระบวนการ การสร้างแบบจำลองทางคณิตศาสตร์ควรเริ่มต้นตามขั้นตอนต่อไปนี้

1. กำหนดขอบเขตของกระบวนการที่ต้องการสร้างแบงจำลองทางคณิตศาสตร์
2. เขียนสมการแสดงแสดงการเปลี่ยนแปลงของ มวลสาร พลังงาน และโมเมนตันที่เกิดภายในกระบวนการ
3. เขียนสมการอื่นเพิ่มเติม เพื่อให้แบบจำลองทางคณิตศาสตร์ของกระบวนการสมบูรณ์ยิ่งขึ้น เช่น สมการแสดงอัตราการทำปฏิกิริยาเคมี สมการแสดงสภาพสมดุลย์ทางเทอร์โมไดนามิกส์
4. พิจารณาสมการสภาวะ ตัวแปรสภาวะ ตัวแปรเข้า และตัวแปรออกเป็นขั้นสุดท้าย

ประโยชน์ของแบบจำลองทางคณิตศาสตร์

การควบคุมกระบวนการอุตสาหกรรมเพื่อรักษาสภาพของกระบวนการให้เป็นไปตามเป้าหมายที่กำหนดไว้ จำเป็นต้องทราบคุณสมบัติของอุปกรณ์ต่าง ๆ ภายในกระบวนการทั้งหมด เพื่อวิเคราะห์คุณสมบัติของระบบควบคุมและผลการควบคุมล่วงหน้าก่อนการก่อสร้างกระบวนการ และติดตั้งอุปกรณ์ควบคุมในกระบวนการ ระบบควบคุมทั่วไปจึงไม่สามารถออกแบบ โดยการทดลองป้อนสัญญาณเข้าให้กระบวนการ เพื่อสังเกตผลการควบคุมที่เกิดขึ้นตามสภาพความจริง ถึงแม้ว่าจะสามารถหาอุปกรณ์ต่าง ๆ ที่ใช้ในกระบวนการจริงได้ทั้งหมดมาใช้ในการทดสอบ

เอกสารนี้ ... ไม่ว่ากรณีใด ๆ ...

ก็ตาม เพราะทำให้เสียเวลาและค่าใช้จ่ายมาก การหาความสัมพันธ์ทางคณิตศาสตร์เพื่ออธิบายปรากฏต่าง ๆ ของกระบวนการหรือการสร้างแบบจำลองทางคณิตศาสตร์ของกระบวนการจึงมีความจำเป็นอย่างยิ่งในการออกแบบระบบควบคุมเพื่อใช้ในการควบคุมกระบวนการอุตสาหกรรม

ระบบทางพลศาสตร์ของขบวนการ
อันดับที่หนึ่ง

ในบทนี้จะกล่าวถึงหลักการสองหลักการคือ หลักการแรกแสดงให้เห็นถึงหลักการเบื้องต้นของการจำลองแบบทางคณิตศาสตร์ของขบวนการ ซึ่งเป็นสิ่งที่จำเป็นโดยเป็นการนำเอาความสัมพันธ์ต่าง ๆ ทางกายภาพ มาเป็นรูปแบบทางคณิตศาสตร์เพื่อที่จะทำการวิเคราะห์ถึงระบบนั้นต่อไป ส่วนหลักการที่สองก็เพื่อที่จะได้ศึกษาถึงความหมายทางกายภาพที่สามารถอธิบายถึงบุคคลิกภาพของขบวนการ

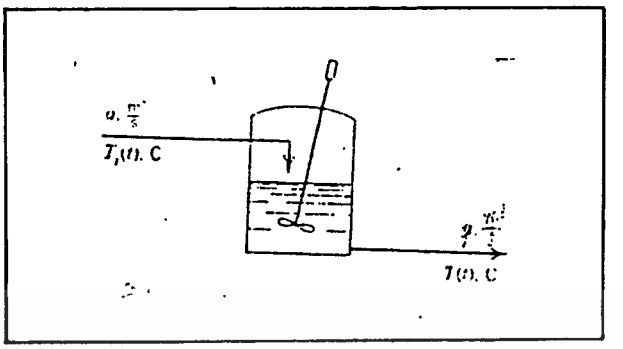
จากที่กล่าวมานี้ทำได้โดยขบวนการตัวอย่างเริ่มจากตัวอย่างที่ง่ายและนำความรู้เหล่านี้มาสู่ขบวนการที่ซับซ้อนมากขึ้น การจำลองรูปแบบของขบวนการทางอุตสาหกรรมจะเริ่มจากกฎของการสมดุลย์ ของมวลหรือพลังงานซึ่งเขียนเป็นความสัมพันธ์ได้คือ

อัตราของมวลหรือพลังงาน - อัตราของมวลหรือพลังงาน = อัตราสะสมของมวลหรือพลังงาน
ที่เข้าไปในขบวนการ ที่ออกจากขบวนการ ที่อยู่ในขบวนการ

ขบวนการความร้อน (Thermal process)

พิจารณาขบวนการในรูป โดยจุดสนใจก็คือ เราพิจารณาผลตอบสนองของอุณหภูมิทางออก T(t) จะมีการเปลี่ยนแปลงเป็นอย่างไรต่อการเปลี่ยนแปลงของอุณหภูมิทางเข้า $T_i(t)$ ซึ่งตัวอย่างนี้จะประกอบด้วย ทางเข้าทางออกของอัตราการไหลแบบปริมาตร ความหนาแน่นของของเหลว ความจุความร้อน ซึ่งของเหลวในถังจะถูกกวน โดยมีฉนวนกั้นความร้อนและเป็นขบวนการแบบ adiabatic สมดุลย์ทางพลังงานที่สภาวะไม่คงที่ (Unsteady-state Energy Balance) ของถังนี้โดยใช้ความสัมพันธ์ระหว่างอุณหภูมิทางเข้าและอุณหภูมิทางออก และจากขบวนการจะได้ความสัมพันธ์ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 1 ขบวนการความร้อน

$$q\rho_i h_i(t) - q\rho h(t) = \frac{d\left(V\rho u(t)\right)}{dt} \qquad (2.1)$$

หรือในเทอมของอุณหภูมิจะได้

$$q\rho C_p T_i(t) - q\rho C_p T(t) = V\rho C_v \frac{dT(t)}{dt}$$

เมื่อ ρ_i, ρ = ความหนาแน่นของของเหลวที่ทางเข้าและทางออกตามลำดับ, Kg/m^3

C_{pi}, C_p = ความจุความร้อนที่แรงดันคงที่ของทางเข้าและทางออกของของเหลวตามลำดับ, J/Kg-C

V = ปริมาตรของของเหลวในถัง, m^3

h_i, h = enthalpies ของของเหลวที่ทางเข้าและทางออกตามลำดับ, J/Kg

u = พลังงานภายในของของเหลวในถัง, J/Kg

C_v = ความจุความร้อนของของเหลวที่ปริมาตรคงที่, J/Kg-C

เมื่อให้ความหนาแน่นและความจุความร้อนเป็นค่าคงที่ที่พิกัดของอุณหภูมิขณะทำงานซึ่งจะเขียนสมการใหม่ได้ว่า

$$q\rho C_p T_i(t) - q\rho C_p T(t) = V\rho C_v \frac{dT(t)}{dt} \qquad (2.2)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สมการ (2.2) ก็คือสมการความแตกต่างแบบเชิงเส้นอันดับที่หนึ่ง ซึ่งจะมีความสัมพันธ์ระหว่างอุณหภูมิทางเข้าและทางออก ซึ่งสมการนี้จะมีตัวไม่รู้ค่าคือ T(t) และ $T_i(t)$ ก็คือตัวแปรทางเข้าฟังก์ชั่นขับ (forcing function) เพื่อขับให้อุณหภูมิทางออกมีการเปลี่ยนแปลงส่วนอุปณหภูมิทางออกจะเรียกว่าตัวแปรทางออกหรือตัวแปรตอบสนอง (Responding Variable) ซึ่งจะมีลักษณะอย่างไรจะขึ้นอยู่กับฟังก์ชั่นขับ

สมดุลย์ของพลังงานที่สภาวะคงที่ (Steady-state energy balance) ก็คือ จุดที่อุณหภูมิทางเข้าและทางออกมีค่าเท่ากันโดยภายในขบวนการไม่มีเทอมของการสะสมหรือเขียนความสัมพันธ์ได้

$$q\rho C_p \overline{T}_i - q\rho C_p \overline{T} = 0 \qquad (2.3)$$

ลบสมการ (2.3) ออกจาก (2.2) จะได้

$$q\rho C_p\left(T_i(t) - \overline{T}_i\right) - q\rho C_p\left(T(t) - \overline{T}\right) = V\rho C_v \frac{d\left(T(t) - \overline{T}\right)}{dt} \qquad (2.4)$$

จากบทที่ 2 ได้กำหนดตัวแปรที่เบี่ยงเบนไว้ดังนี้

$$\mathsf{T}(t) = T(t) - \overline{T} \qquad (2.5)$$

$$\mathsf{T}_i(t) = T_i(t) - \overline{T}_i \qquad (2.6)$$

เมื่อ

$\overline{T}, \overline{T}_i$ = ค่าของอุณหภูมิทางเข้าอุณหภูมิและทางออกตามลำดับที่สภาวะคงที่

$T(t), T_i(t)$ = ตัวแปรที่เบี่ยงเบนของอุณหภูมิทางเข้าและอุณหภูมิทางออกตามลำดับ

แทนสมการ (2.5) และสมการ (2.6) ลงในสมการ (2.4) จะได้

$$q\rho C_p \mathsf{T}_i(t) - q\rho C_p T(t) = V\rho C_v \frac{dT(t)}{dt} \qquad (2.7)$$

สมการ (2.7) คล้ายกับสมการ (2.2) แต่ต่างกันที่เขียนอยู่ในเทอมของอุณหภูมิเบี่ยงเบน การใช้ตัวแปรเบี่ยงเบนในการวิเคราะห์และออกแบบระบบควบคุมขบวนการเป็นสิ่งที่สำคัญมาก ซึ่งตัวแปร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เหล่านี้จะถูกนำไปใช้ในทฤษฎีการควบคุม ดังนั้นความหมายและความสำคัญของตัวแปรเบี่ยงเบนจะต้องทำความเข้าใจให้ดี สมการ (2.7) สามารถที่จะจัดใหม่ได้ดังนี้

$$\frac{V\rho C_v dT(t)}{q\rho C_p dt} + T(t) = T_i(t)$$

และให้

$$\tau = \frac{V\rho C_v}{q\rho C_p} \qquad (2.8)$$

ดังนั้น

$$\tau \frac{dT(t)}{dt} + T(t) = T_i(t) \qquad (2.9)$$

จากสมการความแตกต่างนี้แปลงลาปลาซจะได้

$$\tau sT(s) - \tau T(0) + T(s) = T_i(s)$$

แต่ T(0) = 0

$$T(s) = \frac{1}{\tau s + 1} T_i(s) \qquad (2.10)$$

หรือ

$$\frac{T(s)}{T_i(s)} = \frac{1}{\tau s + 1} \qquad (2.11)$$

สมการ (2.11) เรียกว่าฟังก์ชั่นถ่ายโอนแบบอันดับที่หนึ่ง เนื่องจากถูกพัฒนามาจากสมการความแตกต่างแบบอันดับที่หนึ่ง ขบวนการที่ถูกอธิบายโดยฟังก์ชั่นถ่ายโอนนี้จะเรียกว่า ขบวนการแบบอันดับที่หนึ่ง ระบบอันดับที่หนึ่ง หรือความล้าหลังแบบอันดับที่หนึ่ง บางครั้งอาจจะอ้างถึงระบบแบบความจุเดี่ยว เนื่องจากเป็นฟังก์ชั่นถ่ายโอนที่เหมือนกับระบบทางไฟฟ้าที่มีหนึ่งความต้านทานและหนึ่งความจุ (R-C)

สมมติให้อุณหภูมิเข้าของถังเพิ่มขึ้น A °C ซึ่งเป็นการเปลี่ยนแปลงแบบขั้นมีขนาดเท่ากับ A ทางคณิตศาสตร์เขียนได้ดังนี้

เอกสารนี้สงวนไว้สำหรับใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$q\rho C_p T_i(t) - UA[T(t) - T_s(t)] - q\rho C_p T(t) = V\rho C_v \frac{dT(t)}{dt} \quad (2.14)$$

เมื่อ U = สัมประสิทธิ์ของการส่งผ่านความร้อน, J/m^2-K-s

A = พื้นที่ผิวที่การส่งผ่านความร้อน, m^2

$T_s(t)$ = อุณหภูมิโดยรอบซึ่งเป็นตัวแปรทางเข้า

สัมประสิทธิ์การส่งผ่านความร้อนเป็นฟังก์ชั่นของทุกสิ่งที่เป็นโครงสร้างของถัง อย่างไรก็ตามโดยเฉพาะตัวอย่างนี้กำหนดให้คงที่ ตั้งแต่มวลของของเหลวในถังและความหนาแน่น ความสูงของของเหลว ตลอดจนพื้นที่ผิวที่มีการส่งผ่านความร้อนนั้นคงที่ทั้งหมด เพื่อที่จะหาตัวแปรค่าที่เบี่ยงเบนไปโดยการเริ่มเขียนสมดุลย์ทางพลังงานที่สภาวะคงที่สำหรับขบวนการนี้คือ

$$q\rho C_p \bar{T}_i - UA(\bar{T} - \bar{T}_s) - q\rho C_p \bar{T} = 0 \quad (2.15)$$

ลบสมการ (2.15) ออกจาก (2.14) จะได้

$$q\rho C_p T_i(t) - UA[T(t) - T_s(t)] - q\rho C_p T(t) = V\rho C_v \frac{dT(t)}{dt}$$

$$-q\rho C_p (T(t) - \bar{T}) = V\rho C_v \frac{d(T(t) - \bar{T})}{dt} \quad (2.16)$$

กำหนดให้เป็นค่าของตัวแปรที่เบี่ยงเบน

$$T_s(t) = T_s(t) - \bar{T}_s \quad (2.17)$$

แทน (2.5), (2.6) และ (2.17) ในสมการ (2.16) จะได้

$$q\rho C_p T_i(t) - UA(T(t) - T_s(t)) - q\rho C_p T(t) = V\rho C_v \frac{dT(t)}{dt} \quad (2.18)$$

สมการ (2.18) จะเหมือนกับ (2.14) ยกเว้นจากเราเข้าใจความหมายของตัวแปรที่เบี่ยงเบน สมการ (2.18) เป็นสมการความแตกต่างแบบอันดับที่หนึ่ง ในกรณีนี้จะมีหนึ่งสมการและหนึ่งตัวไม่รู้ค่า T (t) และจะพบตัวแปรอีกตัวหนึ่ง $T_s(t)$ ที่เป็นฟังก์ชั่นขับ เมื่ออุณหภูมิโดยรอบเปลี่ยนแปลงก็จะเป็นเหตุให้เกิดผลต่อการสูญเสียความร้อน และรวมไปถึงอุณหภูมิของของเหลวในขบวนการด้วย สมการ (2.18) สามารถจัดใหม่ได้ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$T_i(t) = \overline{T}_i \qquad t < 0$$
$$T_i(t) = \overline{T}_i + A \qquad t \geq 0$$

หรือที่แสดงในบทที่ 2

$$T_i(t) = Au(t)$$

ทำการแปลงลาปลาซจะได้

$$T_i(s) = \frac{A}{s}$$

แทนลงในสมการ (2.10) จะได้

$$T(s) = \frac{A}{s(\tau s + 1)}$$

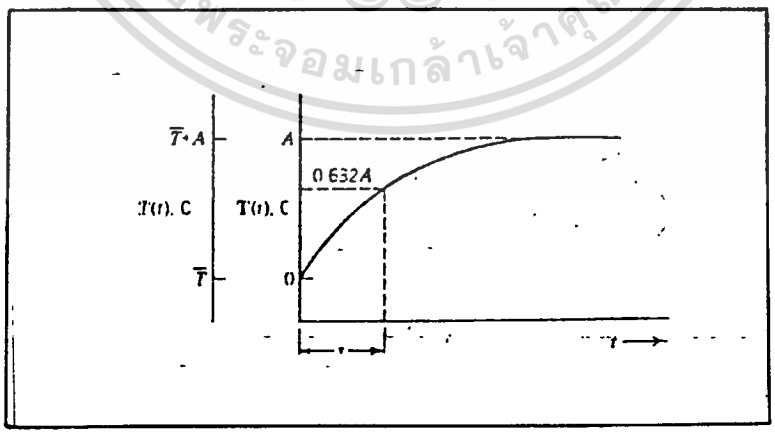
และใช้วิธีการของเศษส่วนย่อยเพื่อการแปลงกลับลาปลาซจะได้

$$T(t) = A\left(1 - e^{-t/\tau}\right) \qquad (2.12)$$

หรือ

$$T(t) = \overline{T} + A\left(1 - e^{-t\tau}\right) \qquad (2.13)$$

สมการ (2.12) และ (2.13) แสดงให้เห็นเป็นกราฟในรูปที่ 2



รูปที่ 2 ผลตอบสนองของขบวนการแบบอันดับที่หนึ่ง
ต่อการเปลี่ยนแปลงของฟังก์ชั่นขับแบบขั้น-เวลาคงตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สมการ (2.12) จะพบกับเครื่องหมายทางกายภาพอย่างหนึ่งคือ ซึ่งถูกเรียกว่า ค่าเวลาคงตัวของขบวนการ สมมติให้ $t = \tau$ จะได้ว่า

$$T(\tau) = A\left(1 - e^{-t/\tau}\right) = A\left(1 - e^{-1}\right)$$
$$T(\tau) = 0.632A$$

นั่นคือในเวลาคงตัวจะเท่ากับจำนวน 63.2% ของการเปลี่ยนแปลงที่เพิ่มขึ้นทั้งหมดของอุณหภูมิที่ได้แสดงในกราฟรูปที่ 2. และค่าเวลาคงตัวก็จะเป็นความสัมพันธ์ความเร็วของผลตอบสนองของขบวนการด้วย ขบวนการที่ตอบสนองช้าต่อฟังก์ชั่นทางเข้าจะมีค่าเวลาคงตังที่มาก และในทำนองเดียวกันขบวนการที่ผลตอบสนองเร็วก็จะมีค่าเวลาคงตัวน้อย หน่วยของเวลาคงตัวจะต้องเป็นหน่วยเวลา จากสมการ (2.8) จะพบว่า

$$\tau = \frac{[m^3][kg/m^3][J/kg-C]}{[m^3/s][kg/m^3][J/kg-C]} = \text{sec}onds$$

เป็นสิ่งที่สำคัญมากที่จะต้องแสดงให้เห็นอย่างจริงจังของค่าเวลาคงตัวที่มีอยู่ในคุณสมบัติทางกายภาพ ซึ่งแตกต่างกันไปรวมทั้งตัวประกอบอื่นของขบวนการที่แสดงในสมการ (.8) จะพบว่าค่าเวลาคงตัวขึ้นอยู่กับปริมาตรของของเหลวในถัง ค่าความจุความร้อน และอัตราไหลในขบวนการ และถ้าคุณลักษณะอันใดอันหนึ่งมีการเปลี่ยนแปลง ค่าเวลาคงตัวก็จะเปลี่ยนแปลงไปด้วย หรือพูดอีกอย่างหนึ่งก็คือ บุคคลิกภาพ ของขบวนการได้เปลี่ยนแปลงไปซึ่งมีผลต่อความเร็วของผลตอบสนอง และที่สำคัญอีกอย่างหนึ่งก็คือค่าของเวลาคงตัวจะคงที่เหนือพิกัดทำงาน ซึ่งเป็นคุณสมบัติของระบบเชิงเส้น แต่จะไม่ได้อยู่ในกรณีนี้ ถ้าระบบไม่ได้เป็นเชิงเส้น ซึ่งจะได้พบต่อไป

ถึงขณะนี้ถังใบนี้ได้ถูกกำหนดให้เป็นฉนวนที่ดี คือจะไม่มีความร้อนสูญเสียไปให้กับบรรยากาศ ซึ่งก็จะไม่มีเทอมของการสูญเสียความร้อน ในสมดุลย์ทางพลังงานจากที่ผ่านมา แต่เราจะไม่เอาตามข้อกำหนดที่แล้ว โดยจะให้มีการสูญเสียความร้อนให้กับบรรยากาศ เพราะฉะนั้นสมดุลย์ทางพลังงานที่สภาวะคงที่คือ

$$q\rho C_p T_i(t) - Q(t) - q\rho C_p T(t) = V\rho C_v \frac{dT(t)}{dt}$$

หรือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\frac{V\rho C_v}{q\rho C_p + UA}\frac{dT(t)}{dt} + T(t) = \frac{q\rho C_p}{q\rho C_p + UA}T_i(t) + \frac{UA}{q\rho C_p + UA}T_s(t) \quad (2.19)$$

หรือ

$$\tau\frac{dT(t)}{dt} + T(t) = K_1 T_i(t) + K_2 T_s(t)$$

เมื่อ

$$\tau = \frac{V\rho C_v}{q\rho C_p + UA} \quad \text{วินาที} \quad (2.20)$$

$$K_1 = \frac{q\rho C_p}{q\rho C_p + UA} \quad \text{ไม่มีหน่วย} \quad (2.21)$$

$$K_2 = \frac{UA}{q\rho C_p + UA} \quad \text{ไม่มีหน่วย} \quad (2.22)$$

ทางด้านขวาของสมการ (2.19) แสดงให้เห็นถึงทั้งสองของฟังก์ชั่น $T_i(t)$ และ $T_s(t)$ และผลตอบสนองจะอยู่ทางด้านซ้ายมือของสมการคือ T(t) ทำการแปลงลาปลาซของสมการ (2.19) จะได้

$$\tau sT(s) - \tau T(0) + T(s) = K_1 T_i(s) + K_2 T_s(s)$$

แต่ T(0) = 0 จัดสมการใหม่จะได้

$$T(s) = \frac{K_1}{\tau s + 1}T_i(s) + \frac{K_2}{\tau s + 1}T_s(s) \quad (2.23)$$

ถ้าอุณหภูมิโดยรอบนั้นยังคงที่ $T_s(t) = \bar{T}_s(t)$ ดังนั้น $T_s(t) = 0$ คือหมายถึงความเบี่ยงเบนนั้นเป็นศูนย์ฟังก์ชั่นถ่ายโอนของความสัมพันธ์ของ อุณหภูมิของขบวนการต่ออุณหภูมิทางเข้าคือ

$$\frac{T(s)}{T_i(s)} = \frac{K_1}{\tau s + 1} \quad (2.24)$$

ถ้าอุณหภูมิทางเข้าของเหลวในขบวนการยังคงที่ ฟังก์ชั่นถ่ายโอนของความสัมพันธ์ของ อุณหภูมิของขบวนการต่ออุณหภูมิโดยรอบคือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\frac{T(s)}{T_s(s)} = \frac{K_2}{\tau s + l} \qquad (2.25)$$

ถ้าทั้งสองของตัวแปรทางเข้าคืออุณหภูมิทางเข้าและอุณหภูมิโดยรอบมีการเปลี่ยนแปลงแล้ว จะใช้วิธีการของหาผลตอบสนองที่ละส่วนของตัวแปรทางเข้าและนำมารวมกัน (Superposition)

ในสามสมการสุดท้าย จะพบตัวประกอบที่สำคัญคือ K ซึ่งถูกเรียกว่าอัตราการขยายของขบวน (Process อัตราการขยาย) หรือ อัตราการขยายที่สภาวะคงที่ (Steady-state อัตราการขยาย) เพื่อบ่งบอกถึงคุณลักษณะทางกายภาพของอัตราการขยาย โดยการสมมติให้อุณหภูมิทางเข้าของถัง นั้นเพิ่มขึ้นขนาด A °C โดยการเปลี่ยนแปลงขั้นและผลตอบสนองของอุณหภูมิต่อฟังก์ชั่นขับอันนี้คือ

$$T(s) = \frac{K_l}{s(\tau s + l)}$$

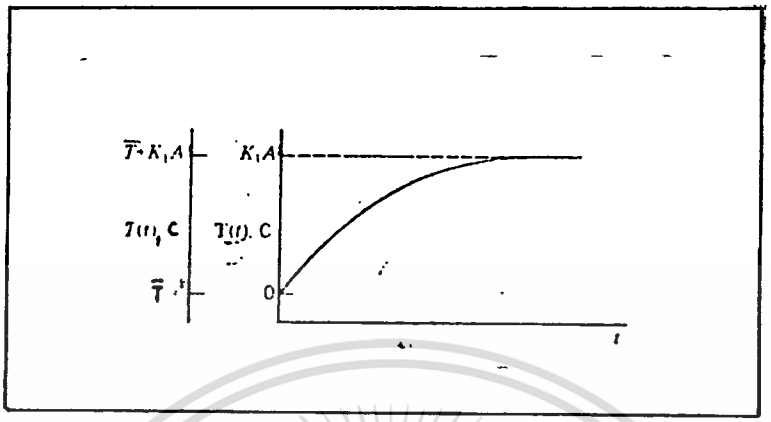
$$T(t) = K_l A\left(l - e^{-t/\tau}\right) \qquad (2.26)$$

$$T(t) = \overline{T} + K_l A\left(l - e^{-t/\tau}\right) \qquad (2.27)$$

ผลตอบสนองแสดงในรูป3จำนวนทั้งหมดที่เปลี่ยนแปลงมีค่าเท่ากับ KA คือค่าของอัตราการขยายคูณกับค่าของการเปลี่ยนแปลงของฟังก์ชั่นขับ เพราะฉะนั้นอัตราการขยายจะสามารถบอกเราได้ว่า ตัวแปรทางออกจะมีการเปลี่ยนแปลงมากเท่าใดต่อการเปลี่ยนแปลงต่อหน่วยของฟังก์ชั่นขับหรือตัวแปรทางเข้า นั่นคือ อัตราการขยายจะถูกกำหนดให้เป็นความไว (Senditivity) ของขบวนการ

$$K\frac{\Delta 0}{\Delta l}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3 แสดงผลตอบสนองของขบวนการแบบอันดับที่หนึ่ง
ต่อการเปลี่ยนแปลงแบบขั้นของฟังก์ชั่นขับ

อัตราการขยายจะขึ้นอยู่กับคุณสมบัติทางกายภาพ หรือบุคคลิกภาพของขบวนการภายใต้การควบคุมดังแสดงในสมการ (2.21) และ (2.23) ซึ่งอัตราการขยายของขบวนการอันนี้จะขึ้นอยู่กับอัตราการไหล ความหนาแน่น ความจุความร้อนของขบวนการ สัมประสิทธิ์ของการส่งผ่านความร้อนและพื้นที่ผิวในการส่งผ่านความร้อน ถ้าสิ่งหนึ่งสิ่งใดมีการเปลี่ยนแปลงก็จะทำให้คุณสมบัติของขบวนการมีการเปลี่ยนแปลงด้วย ซึ่งจะมีผลต่อ อัตราการขยาย และส่งผ่านไปถึงค่าเวลาคงตัว ในตัวอย่างมีจะ อัตราการขยายอยู่ 2 ค่า คือ K_2 โดย K_1 ได้จากความสัมพันธ์ของอุณหภูมิทางออกต่ออุณหภูมิทางเข้า ส่วน K_2 ได้จากความสัมพันธ์ของอุณหภูมิทางออกต่ออุณหภูมิโดยรอบ หน่วยของอัตราการขยาย จะต้องเป็นหน่วยของตัวแปรทางออกหารด้วยตัวแปรทางเข้าหรือฟังก์ชั่นขับและในความเป็นจริง อัตราการขยาย จะต้องเป็นค่าบวกเสมอ สมการ (2.23) แสดงว่ามีเพียงค่าเวลาคงตัวเพียงค่าเดียวในขบวนการ นั้นคือเวลาที่ทำให้ อุณหภูมิทางออกนั้นเพิ่มขึ้นสู่ขอบเขตของการเปลี่ยนแปลงทั้งหมด ที่เป็นผลมาจากอุณหภูมิทางเข้า และจะเท่า

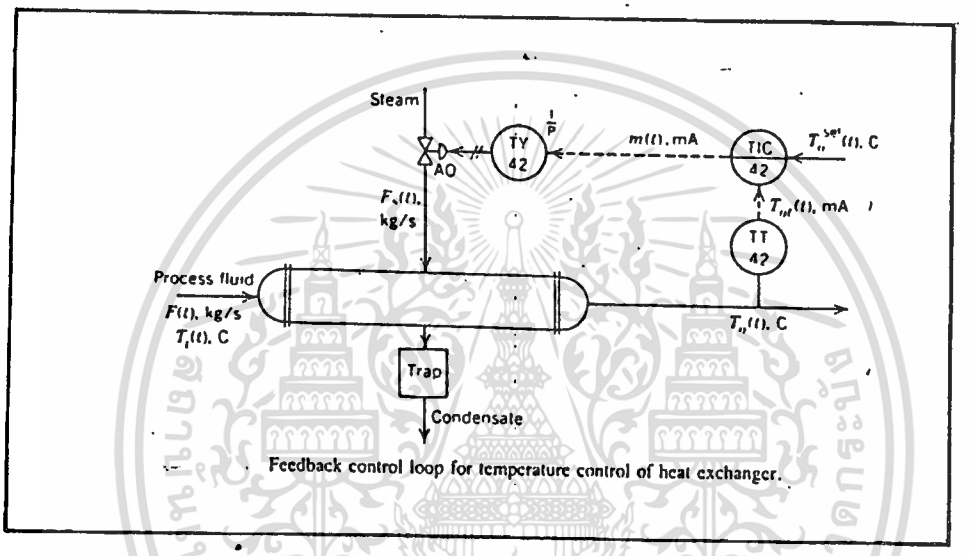
ระบบควบคุมแบบป้อนกลับ

หลักการของการควบคุมแบบป้อนกลับแม้ว่าเกิดขึ้นมานานกว่า 200 ปี แล้วก็ตาม แต่ก็ไม่ได้ถูกนำไปใช้ในอุตสาหกรรมจนกระทั่ง James Watt ได้ทำการสร้างเครื่องปรับความเร็วให้กับเครื่องจักรไอน้ำของเขาจากนั้นอุตสาหกรรมมากมายได้หันมาใช้หลักการของการควบคุมแบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ป้อนกลับและเป็นที่แพร่หลายจนกระทั่งทุกวันนี้ ต่อมาได้มีการพัฒนาระบบควบคุมแบบป้อนกลับขั้นสูงเมื่อ 50 ปีที่แล้วมา เพื่อให้สมรรถนะที่ดีกว่า และสามารถนำมาใช้ทดแทนระบบควบคุมแบบป้อนกลับแบบเดิม ซึ่งเราจะได้ศึกษาวิธีการขั้นสูงนี้ในบทเรียนต่อมา

เพื่อแสดงหลักการของการควบคุมแบบป้อนกลับมาพิจารณาตัวอย่างของกระบวนการแลกเปลี่ยนความร้อนในบทที่ 1 ที่แสดงในรูปที่ 4



รูปที่ 4 วงรอบควบคุมแบบป้อนกลับสำหรับการควบคุมอุณหภูมิของกระบวนการแลกเปลี่ยนความร้อน

จุดประสงค์ก็เพื่อที่จะรักษาค่าของอุณหภูมิทางออกของของไหล $T_o(t)$ ที่ต้องการหรือจุดเป้าหมาย $T_o^{set}(t)$ ไม่ว่าอัตรการไหล F(t) และอุณหภูมิทางเข้า $T_i(t)$ ของของไหลจะมีการแปรค่าไปเป็นอย่างไรก็ตาม ส่วนอัตราการไหลของน้ำ $F_s(t)$ เป็นตัวแปรที่สามารถปรับค่าเพื่อใช้ในการควบคุมอุณหภูมิทางออกของของไหล โดยพิจารณาจากจำนวนพลังงานที่จ่ายให้กับของไหล

จะเห็นได้ว่าอุณหภูมิทางออกหรือตัวแปรที่ถูกควบคุมได้ถูกวัดด้วยอุปกรณ์ตรวจจับ (TT42) และสร้างสัญญาน $T_{ot}(t)$ ที่เป็นสัดส่วนกับอุณหภูมิ อุปกรณ์ส่งสัญญานจะส่งสัญญานดังกล่าวให้กับเครื่องควบคุม (TIC42) เมื่อทำการเปรียบเทียบค่ากับจุดเป้าหมายฟังก์ชั่นของเครื่องควบคุมก็ทำการสร้างสัญญานตัวแปรจากการจัดการ m(t) ตามค่าผิดพลาดหรือค่าความแตกต่างระหว่าง ค่าที่วัดได้กับค่าที่จุดเป้าหมายที่ทางออกของเครื่องควบคุมจะถูกต่ออยู่กับอุปกรณ์บังคับวาล์วควบคุมไอน้ำโดยผ่านอุปกรณ์แปลงกระแสไฟฟ้าเป็นแรงดันลม (I/P) ทั้งนี้เพราะว่าใน

เอกสารนี้เป็นลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ห้ามนำไปใช้เพื่อการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

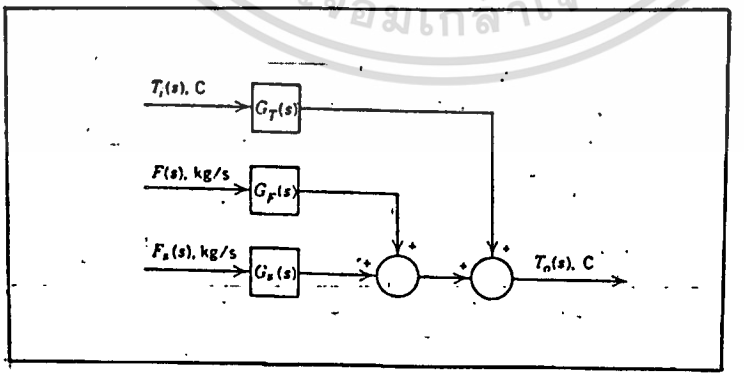
ตัวอย่างนี้อุปกรณ์ส่งสัญญานและเครื่องควบคุมต่างก็เป็นแบบอิเลคทรอนิคส์แต่อุปกรณ์บังคับวาล์วควบคุมทำงานโดยใช้แรงดันลม โดยฟังก์ชั่นของอุปกรณ์บังคับวาล์วควบคุมจะมีตำแหน่งที่เป็นสัดส่วนกับสัญญานทางออกของเครื่องควบคุมและเป็นผลให้อัตราไหลของไอน้ำเป็นฟังก์ชั่นกับตำแหน่งของวาล์วควบคุม

เทอม "Feedback" จำกัดความจากตัวแปรที่ถูกควบคุมถูกวัด และผลของการวัดนี้ได้ถูกป้อนกลับมาเพื่อที่ปรับเปลี่ยนตำแหน่งของวาล์วควบคุมไอน้ำให้เหมาะสมกับอุณหภูมิทางออกที่วัดได้ ในการวิเคราะห์ที่ดีที่สุดของการทำงานในวงรอบการควบคุมโดยการเริ่มเขียนบล๊อคไดอะแกรมของแต่ละส่วนของอุปกรณ์และต่อเชื่อมทางออกของสัญญานเข้ากับทางเข้าของสัญญานของอีกส่วนหนึ่งจากรูปที่ 5 กระบวนการแลกเปลี่ยนความร้อนประกอบด้วยสามบล๊อคแต่ละบล๊อคมีหนึ่งทางเข้าของสัญญาน

$G_T(s)$ เป็นฟังก์ชั่นถ่ายโอนที่เป็นความสัมพันธ์ระหว่างอุณหภูมิทางออกต่ออุณหภูมิทางเข้ามีหน่วยเป็น C/C

$G_F(s)$ เป็นฟังก์ชั่นถ่ายโอนที่เป็นความสัมพันธ์ระหว่างอุณหภูมิทางออกต่ออัตราไหลของของไหลมีหน่วยเป็น C/(Kg/s)

$G_S(s)$ เป็นฟังก์ชั่นถ่ายโอนที่เป็นความสัมพันธ์ระหว่างอุณหภูมิทางออกต่ออัตราไหลของไอน้ำมีหน่วยเป็น C/(Kg/s)



รูปที่ 5 บล๊อคไดอะแกรมของกระบวนการแลกเปลี่ยนความร้อนในรูปที่ 6-1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บล๊อคไดอะแกรมที่สมบูรณ์สำหรับวงรอบควบคุมแบบป้อนกลับแสดงในรูปที่ 6-3 โดยมีตัวแปรดังนี้

E(s) เป็นสัญญานค่าผิดพลาด, mA

$G_c(s)$ เป็นฟังก์ชั่นถ่ายโอนของเครื่องควบคุม, mA/mA

$G_V(s)$ เป็นฟังก์ชั่นถ่ายโอนของวาล์วควบคุม, (kg/s)mA

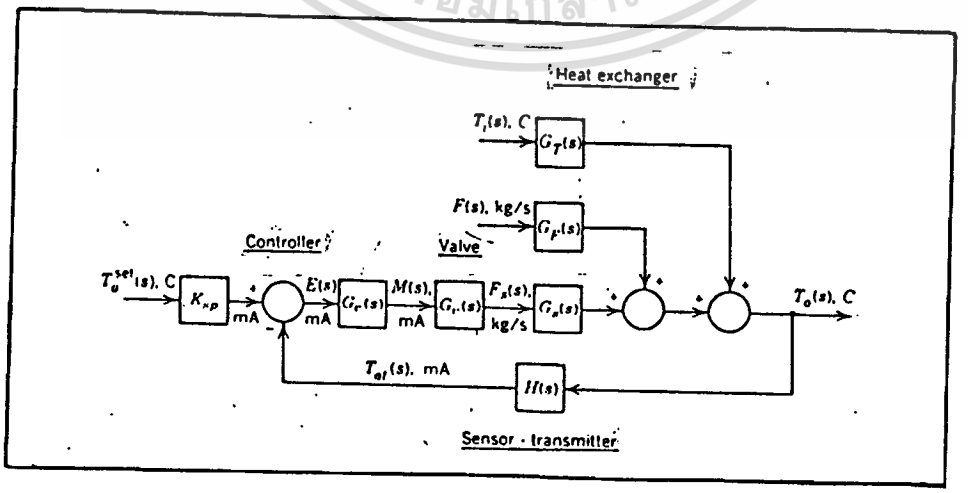
H(s) เป็นฟังก์ชั่นถ่ายโอนของอุปกรณ์ตรวจจับอุปกรณ์ส่งสัญญาน, mA/C

K_{sp} เป็นตัวแปรประกอบสเกล (Scale Factor) ของอุณหภูมิจุดเป้าหมาย, mA/C

อัตราการขยายคงที่ของอุปกรณ์แปลงกระแสไฟฟ้าเป็นแรงดันลม (I/P ในรูปที่ 6-1) ที่รวมอยู่ในฟังก์ชั่นถ่ายโอนของวาล์วควบคุม, $G_V(s)$ ซึ่งอัตราการขยายของอุปกรณ์แปลงดังกล่าวคือ

$$\frac{\Delta P}{\Delta I} = \frac{(15-3)\,psi}{(20-4)\,mA} = 0.75\,psi/mA$$

ดังนั้นหน่วยของ $G_V(s)$ คือ (kg/s)mA โดยสมมติว่าแรงดันตกคร่อมที่วาล์วควบคุมไอน้ำคงที่ส่วนเทอม K_{sp} ที่สัญญานค่าจุดเป้าหมายจะถูกรวมกับค่าปรับแปลงสเกล (Scale Conversion) โดยทั่วไปจะทำการปรับแต่งให้มีหน่วยเหมือนกับตัวแปรที่ถูกควบคุมและเหมือนกับอุปกรณ์ส่งสัญญาน นั่นคือ องศาเซลเซียสต่อมิลลิแอมป์, C/mA เมื่อเครื่องควบคุมได้รับค่าจากการวัดและค่าที่จุดเป้าหมายอยู่ในสเกลเดียวกันแล้วก็แสดงว่า K_{sp} คือตัวเลขที่เท่ากับค่าอัตราขยายของอุปกรณ์ส่งสัญญานที่สภาวะคงที่



รูปที่6 บล๊อคไดอะแกรมวงรอบการควบคุมอุณหภูมิของกระบวนการแลกเปลี่ยนความร้อน
(บล๊อคอุปกรณ์ I/P ได้ถูกรวมกับบล๊อคของวาล์วควบคุมแล้ว)

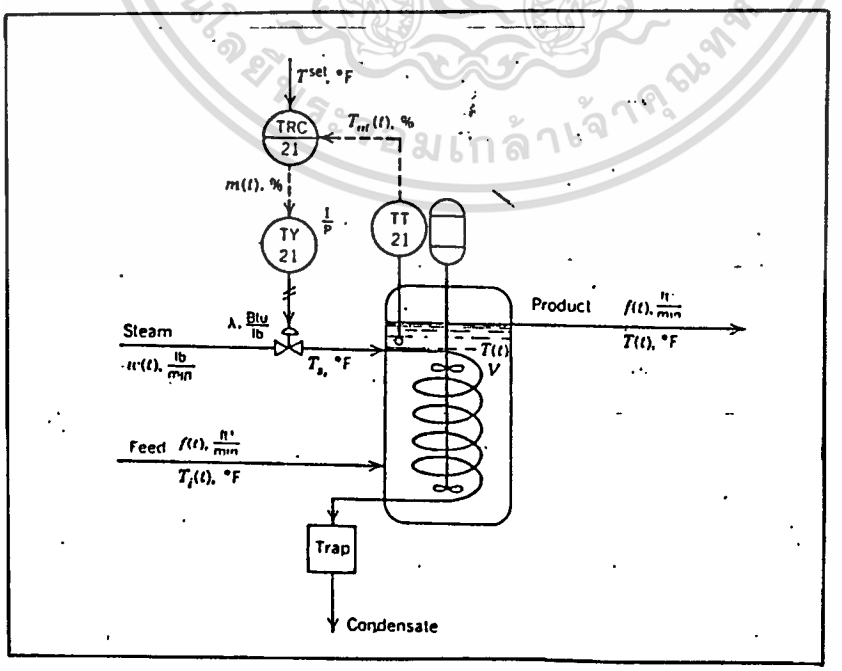
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กระบวนการที่เลือกใช้ในการจำลอง

ระบบควบคุมอุณหภูมิสำหรับถังกวนให้ความร้อนแบบต่อเนื่อง ที่จะแสดงรายละเอียดดังต่อไปนี้ เป็นขบวนการที่ถูกเลือกมาทำการจำลอง โดยใช้หลักการพื้นฐานทางวิศวกรรมกระบวนการมาใช้ โดยจะแบ่งรายละเอียดเป็นหัวข้อดังนี้

1. กระบวนการ
2. คอยล์ประกอบด้วย Coil
3. สภาวะในการออกแบบ
4. อุปกรณ์วัด-อุปกรณ์ส่งสัญญานของอุณหภูมิ
5. วาล์วควบคุม

ระบบควบคุมอุณหภูมิของถังกวนให้ความร้อนแบบต่อเนื่อง ถังกวนถูกแสดงในรูป ถูกให้ความร้อนแก่กระบวนการโดยไอน้ำ เพื่อที่จะกวนผสมกันให้เกิดการขึ้นรูปของของเหลวที่เป็นผลิตภัณฑ์ ในการควบคุมอุณหภูมิเป็นสิ่งที่สำคัญ เพราะว่าอุณหภูมิที่สูงเกินไปจะทำให้เรียงตัวกันไม่ดีของผลิตภัณฑ์ และขณะที่อุณหภูมิที่ต่ำก็จะทำให้การผสมเข้ากันไม่ดีเช่นกัน ถังดังกล่าวถูกให้ความร้อนโดยไอน้ำที่ควบแน่นภายในคอยล์ เครื่องควบคุมที่ใช้เป็นแบบ PID ถูกใช้ในการควบคุมอุณหภูมิในถังโดยจัดการกับตำแหน่งของวาล์วไอน้ำให้เหมาะสม ในที่นี้ต้องการจะหาบล๊อคไดอะแกรมที่สมบูรณ์และสมการคุณลักษณะของวงรอบจากข้อมูลสำหรับการออกแบบดังต่อไปนี้



รูป 7 ระบบควบคุมอุณหภูมิสำหรับถังกวนให้ความร้อนแบบต่อเนื่อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. กระบวนการ

วัตถุดิบ

— มีความหนาแน่น ρ เป็น $68.0\ lb/ft^3$

— ความจุความร้อน C_p เป็น $0.80\ BTU / lb^{\circ}F$

— ปริมาตรของเหลวในถังปฏิกิริยาถูกรักษาให้คงที่ $120\ ft^3$

2. **คอยล์ประกอบด้วย**

— ความยาว *205 ft 4 inch*

— ท่อเหล็กเบอร์ *40* น้ำหนัก *108 lb/ft*

— ความจุความร้อน $0.12\ BTU / lb^{\circ}F$

— เส้นผ่าศูนย์กลาง *4.500 inch*

— สัมประสิทธิ์การส่งผ่านความร้อนทั้งหมด ขึ้นอยู่กับพื้นที่ผิวด้านนอกของคอยล์โดยประมาณ $2.1\ BTU/min\ ft^2\ ^{\circ}F$

— ไอน้ำมีค่าอิ่มตัวที่แรงดัน 30 psia สามารถที่จะกำหนดให้มีความร้อนแฝงของการควบแน่น λ มีค่าคงที่ *966 BTU / lb*

3. **สภาวะในการออกแบบ**

— อัตราไหลของวัตถุดิบ *f* ที่สภาวะการออกแบบเป็น $15\ ft^3 / min$

— อุณหภูมิทางเข้า T_i เป็น $100\ ^{\circ}F$

สิ่งที่อยู่ภายในถังจะต้องถูกรักษาอุณหภูมิ *T* ไว้ใน $150\ ^{\circ}F$ และเป็นไปได้ที่การเปลี่ยนแปลงของอัตราการไหลของวัตถุดิบและอุณหภูมิที่ทางเข้าของวัตถุดิบจะเป็นตัวแปรรบกวนของระบบ

อุปกรณ์วัด-อุปกรณ์ส่งสัญญาณของอุณหภูมิ

— อุปกรณ์วัดอุณหภูมิได้ถูกปรับเทียบที่พิกัด 100 - 200 องศาฟาเรนต์ไฮต์ และค่าเวลาคงตัว เป็น 0.75 min -

วาล์วควบคุม

— วาล์วควบคุมได้ถูกออกแบบสำหรับ 100% ที่เกินความสามารถและแรงดันตกที่แปรค่าไปสามารถตัดทิ้งได้

— วาล์วที่ใช้เป็นแบบอีควอลเปอร์เซนต์เทจด้วยพารามิเตอร์ของความเป็นพิกัด α เป็น 50

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

— อุปกรณ์ขับวาล์วที่ค่าเวลาคงตัว τ_v เป็น 0.20 min

วิธีทำ จุดมุ่งหมายของเราจะต้องดำเนินการทางสมการเพื่อที่จะอธิบายพฤติกรรมทางพลศาสตร์ของถัง วาล์วควบคุม อุปกรณ์วัด-ส่งสัญญาน และเครื่องควบคุม จากนั้น เราจะปรับเป็นเชิงเส้นแปลงลาปลาซ เพื่อที่จะหาบล๊อคไดอะแกรมของวงรอบต่อไป

สำหรับกระบวนการ

กำหนดให้ความร้อนสูญเสียน้อยมาก ส่วนการผสมนั้นเป็นไปอย่างสมบูรณ์และปริมาตรและคุณสมบัติทางกายภาพนั้น ยังคงที่ สมดุลย์ทางพลังงานของของเหลวในถังเป็นไปตามสมการดังต่อไปนี้

$$\rho C_p \frac{dT(t)}{dt} = f(t)\rho C_p T_i(t) + UA[T_s(t) - T(t)] - f(t)\rho C_p T(t)$$

เมื่อ

A เป็นพื้นที่ในการส่งผ่านความร้อน, ft^2

$T_s(t)$ เป็นอุณหภูมิของไอน้ำที่ควบแน่น, องศาฟาเรนต์ไฮต์

และสัญลักษณ์อื่นได้ถูกกำหนดในลักษณะของปัญหาสำหรับของเหลวที่อยู่ในถังค่า C_v ในเทอมของการสะสมได้ประมาณให้มีค่าเท่ากับ C_p

กำหนดให้โลหะของคอยล์จำเป็นอย่างยิ่งที่จะต้องเป็นอุณหภูมิเดียวกันในขณะที่ควบแน่นสมดุลย์ทางพลังงานที่คอยล์จะได้ดังนี้

$$C_m \frac{dT_s(t)}{dt} = w(t)\lambda - UA[T_s(t) - T(t)]$$

เมื่อ

$W(t)$ เป็นอัตราไหลของน้ำไอน้ำ, lb/min

C_M เป็นความจุความร้อนของโลหะของคอยล์, BTU/F

อัตราไหลของไอน้ำ คือทางออกของวาล์วควบคุมและเป็นทางเข้าของกระบวนการและกระบวนการของเราก็ได้ถูกจำลองแบบโดยสมบูรณ์แล้ว

เอกสารนี้มีไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับวาล์วควบคุม

สมการของวาล์วแบบอีควอลเปอร์เซนต์เทจด้วยแรงดันทางด้านสูงคงที่และแรงดันตกที่คงที่สามารถที่จะเขียนได้ดังนี้

$$W(t) = W_{max}\alpha^{vp(t)H}$$

เมื่อ

W_{max} เป็นอัตราไหลสูงสุดที่ไหลผ่านวาล์ว, lb/min

α เป็นพารามิเตอร์ของความเป็นพิกัดของวาล์วแบบอีควอลเปอร์เซนต์เทจ

$vp(t)$ ตำแหน่งของวาล์วในสเกล 0 - 1

การแปรค่าของแรงดันที่ตกคร่อมวาล์วกับอุณหภูมิของไอน้ำที่ควบแน่น (และความดัน) ได้ถูกตัดทิ้งไปในตัวอย่างแบบง่ายนี้ ส่วนอุปกรณ์ขับวาล์วสามารถที่จะจำลองแบบโดยแทนด้วยระบบอันดับที่หนึ่ง

$$P(s) = \frac{1/100}{\tau_v s + 1} M(s)$$

เมื่อ $M(s)$ เป็นสัญญานทางออกของเครื่องควบคุมเป็นเปอร์เซนต์

สำหรับอุปกรณ์วัด-ส่งสัญญาน *(TT21)*

อุปกรณ์วัด-ส่งสัญญาน สามารถที่จะแสดงโดยระบบอันดับที่หนึ่ง

$$\frac{T_{ot}(s)}{T(s)} = \frac{K_t}{\tau_t s + 1}$$

เมื่อ

$T_{ot}(s)$ เป็นการแปลงลาปลาซของสัญญานทางออกของอุปกรณ์ส่งสัญญาน, %

เครื่องควบคุมแบบป้อนกลับ *(TRC21)*

ฟังก์ชั่นถ่ายโอนของเครื่องควบคุมแบบ PID

$$G_c(s) = K_c 1 + \frac{1}{\tau_I s} + \tau_D s = \frac{M(s)}{R(s) - T_{ot}(s)}$$

เมื่อ K_c = เป็นอัตราขยายของเครื่องควบคุม, τ_I = เป็นเวลาอินทริกัล, τ_D = เป็นเวลาเดอริเวทีฟ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สมการสำหรับวงรอบการควบคุมอุณหภูมินั้นเสร็จสมบูรณ์แล้ว ขั้นตอนต่อไปเราจะทำการปรับเป็นเชิงเส้นของสมการที่ได้จำลองแบบและแปลงลาปลาซเพื่อหาบล๊อคไดอะแกรมของวงรอบ

ปรับเป็นเชิงเส้นและแปลงลาปลาซ

โดยวิธีการที่ได้ศึกษาไปแล้วในส่วนที่ (2-3) เราปรับเป็นเชิงเส้นของสมการในเทอมของตัวแปรที่เบี่ยงเบน

$$\rho C_p \frac{dT(t)}{dt} = \bar{f}\rho C_p T_i(t) + \rho C_p\left(\overline{T}_i - \overline{T}\right)F(t) + UAT_s(t) - \left(UA + \bar{f}\rho C_p\right)T(t)$$

$$C_m \frac{dT_s(t)}{dt} = \lambda W(t) - UAT_s(t) + UAT(t)$$

เมื่อ $T(t), T_s(t), F(t), T_i(t), W(t)$ เป็นตัวแปรเบี่ยงเบนจากนั้นแปลงลาปลาซและจัดรูปใหม่จะได้

$$T(s) = \frac{K_F}{\tau s + 1}F(s) + \frac{K_i}{\tau s + 1}T_i(s) + \frac{K_s}{\tau s + 1}T_s(s)$$

$$T_s(s) = \frac{1}{\tau_c s + 1}T(s) + \frac{K_W}{\tau_c s + 1}W(s)$$

เมื่อ

$$\tau = \frac{V\rho C_p}{UA + \bar{f}\rho C_p} \qquad \tau_c = \frac{C_M}{UA}$$

$$K_F = \frac{\rho C_p\left(\overline{T}_i - \overline{T}\right)}{UA + \bar{f}\rho C_p} \qquad K_i = \frac{\bar{f}\rho C_p}{UA + \bar{f}\rho C_p}$$

$$K_s = \frac{UA}{UA + \bar{f}\rho C_p} \qquad K_W = \frac{\lambda}{UA}$$

ผลของการปรับเป็นเชิงเส้นที่สมการของวาล์วคือ

$$(t) = W_{\max}(\ln\alpha)\alpha^{\bar{v}p-1}VP(t)$$
$$= \overline{W}(ln\alpha)VP(t)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
037158

เมื่อ *VP(t)* เป็นตัวแปรเบี่ยงเบนของตำแหน่งวาล์วแปลงลาปลาซจะได้

$$(s) = \overline{W}(ln\alpha)VP(s)$$

รวมสมการนี้กับฟังก์ชั่นถ่ายโอนของอุปกรณ์ขับวาล์วเราสามารถที่จะตัดเทอม *VP(s)* ออกไปได้

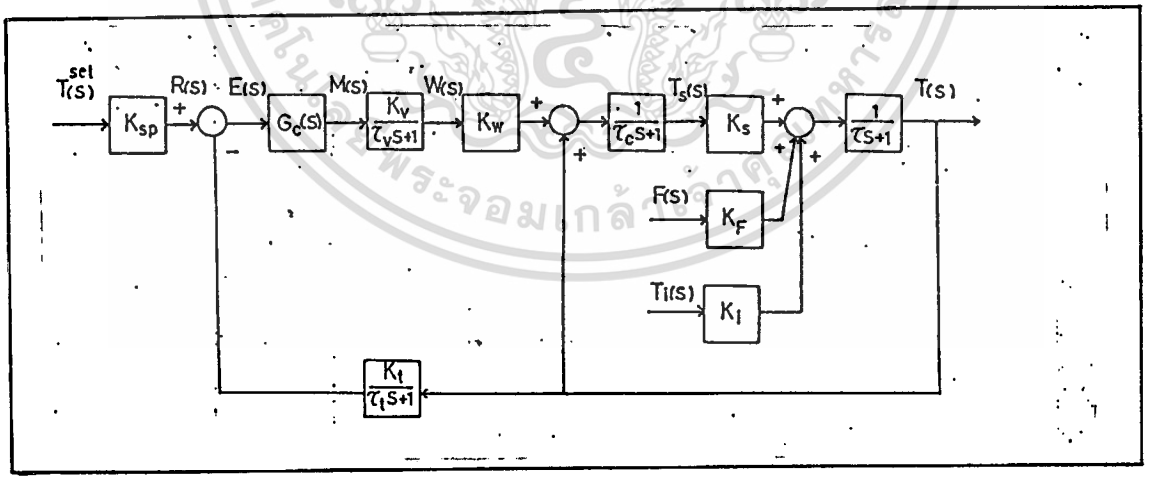
$$\frac{W(s)}{M(s)} = \frac{K_v}{\tau_v s + 1}$$

เมื่อ

$$K_v = \frac{\overline{W}(ln\alpha)}{100}$$

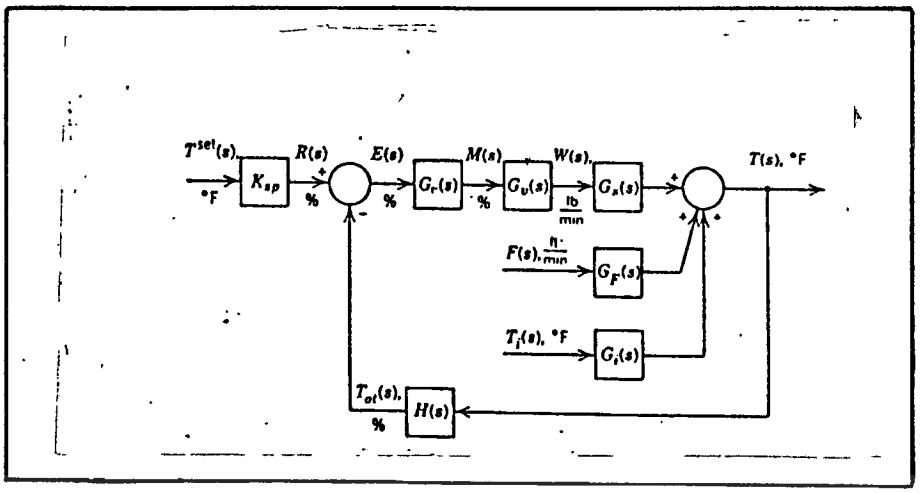
จากที่ได้เคยศึกษาในบทที่ 5 เราสามารถที่จะหาอัตราขยายของอุปกรณ์ส่งสัญญาณได้โดย

$$K_T = \frac{100-0}{200-100} = 1.0\%/^{\circ}F$$



รูปที่ 8 แสดงบล๊อคไดอะแกรมของวงรอบการควบคุมอุณหภูมิ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 9 แสดงบล๊อคไดอะแกรมที่ง่ายขึ้นของวงรอบการควบคุมอุณหภูมิ

วงรอบแสดงในรูปที่ 8 ฟังก์ชั่นถ่ายโอนทั้งหมดในไดอะแกรมนี้ ได้ถูกพัฒนามาจากสมการก่อนหน้านี้ทั้งสิ้น โดยใช้กฎเกณฑ์ของบล๊อคไดอะแกรม เราสามารถที่จะทำให้ง่ายเข้าดังที่แสดงในรูปที่ 9 และฟังก์ชั่นถ่ายโอนที่แสดงในไดอะแกรมคือ

$$G_F(s)=\frac{K_F(\tau_c s+1)}{(\tau s+1)(\tau_c s+1)-K_s}$$

$$G_i(s)=\frac{K_i(\tau_c s+1)}{(\tau s+1)(\tau_c s+1)-K_s}$$

$$G_s(s)=\frac{K_W K_s}{(\tau s+1)(\tau_c s+1)-K_s}$$

ฟังก์ชั่นถ่ายโอนแบบวงรอบปิดของแต่ละทางเข้าคือ

$$\frac{T(s)}{T^{set}(s)}=\frac{K_{sp}G_c(s)G_v(s)G_s(s)}{1+H(s)G_c(s)G_v(s)G_s(s)}$$

$$\frac{T(s)}{F(s)}=\frac{G_F(s)}{1+H(s)G_c(s)G_v(s)G_s(s)}$$

$$\frac{T(s)}{T_i(s)}=\frac{G_i(s)}{1+H(s)G_c(s)G_v(s)G_s(s)}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อ $G_v(s) = \frac{K_v}{\tau_v s + 1}$

$$H(s) = \frac{K_i}{\tau_i s + 1}$$

สมการคุณลักษณะของวงรอบคือ

$$1 + \frac{K_t}{\tau_t s + 1} K_c \left[1 + \frac{1}{\tau_i s} + \tau_D s \right] \frac{K_v}{\tau_v s + 1} \frac{K_W K_s}{(\tau_v s + 1)(\tau_c s + 1) - K_s} = 0$$

ค่าเชิงเลขสามารถหาได้ดังนี้

$K_{sp} = K_t = 1.0\%/^{\circ}F$ $\tau_t = 0.75\ min$ $\tau_v = 0.20\ min$

จากข้อกำหนดของคอยล์จะได้

$$A = (205\,ft)\pi\left(\frac{4.500}{12}\right)ft = 241.5\,ft^2$$

$$C_M = (205\,ft)\left(10.8\frac{lb}{ft}\right)\left(0.12\frac{But}{lb^{\circ}F}\right) = 265.7\,Btu/^{\circ}F$$

$$\tau = \frac{(120)(68.0)(0.80)}{(2.1)(241.5) + (15)(68)(0.80)} = 4.93\,min$$

$$\tau = \frac{265.7}{(2.1)(241.5)} = 0.524\,min$$

$$K_F = \frac{(68)(0.80)(100-150)}{(2.1)(241.5) + (15)(68)(0.80)} = -2.06^{\circ}F/\left(ft^3/min\right)$$

$$K_i = \frac{(15)(68)(0.80)}{(2.1)(241.5) + (15)(68)(0.80)} = 0.617^{\circ}F/^{\circ}F$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$K_s = \frac{(2.1)(241.5)}{(2.1)(241.5)+(15)(68)(0.80)} = 0.383^\circ F/^\circ F$$

$$K_W = \frac{966}{(2.1)(241.5)} = 1.905^\circ F/(lb/min)$$

เพื่อที่จะหาขนาดของวาล์วควบคุม เราจะใช้เงื่อนไขการออกแบบที่สภาวะคงที่

$$\bar{f}\rho C_p \bar{T}_i + UA(\bar{T}_s - \bar{T}) - \bar{f}\rho C_p \bar{T} = 0$$

$$\overline{W}\lambda - UA(\bar{T}_s - \bar{T}) = 0$$

$$\bar{T}_s = \frac{(15)(68)(0.80)(150-100)}{(2.1)(241.5)} + 150 = 230^\circ F$$

$$\overline{W} = \frac{(2.1)(241.5)(230-150)}{966} = 42.2lb/min$$

$$K_v = \frac{(42.2)(\ln 50)}{100} = 1.652lbm/min-\%$$

$$W_{max} = 2\overline{W} = 84.4lb/min$$

ด้วยค่าที่เป็นเชิงเลขเหล่านี้ สมการคุณลักษณะคือ

$$s(0.75s+1)(0.20s+1)[(4.94s+1)(0.524s+1)-0.383]$$

$$+(1.0)K_c\left[s+\frac{1}{\tau_1}+\tau_D s^2\right](1.652)(1.905)(0.383)=0$$

หรือ

$$0.387s^5 + 3.272s^4 7.895s^3 + (6.043 + 1.205K_c\tau_D)s^2$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$+\left(0.617+1.205K_c\right)s+1.205K_c/\tau_I=0$$

ตัวอย่างนี้แสดงให้เห็นถึงหลักการพื้นฐานทางวิศวกรรมกระบวนการที่สามารถนำไปใช้ในงานเพื่อที่จะวิเคราะห์ถึงวงรอบการควบคุมแบบป้อนกลับที่ง่ายได้ จากสมการคุณลักษณะเราสามารถศึกษาถึงเสถียรภาพของวงรอบหรือจากฟังก์ชั่นถ่ายโอนแบบวงรอบปิดได้ เราสามารถที่จะคำนวณผลตอบสนองของวงรอบปิดต่อการแปรค่าไปของทางเข้าที่เป็นฟังก์ชั่นขับหรือค่าที่แตกต่างกันไปของพารามิเตอร์ที่ถูกปรับเปลี่ยนไปเครื่องควบคุม ได้เช่นเดียวกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
แนวความคิดในการสร้างโปรแกรมการคำนวณ เพื่อใช้ในการจำลองกระบวนการทางความร้อน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

แนวความคิดในการสร้างโปรแกรมการคำนวณ เพื่อใช้ในการจำลองกระบวนการ

ขบวนการแลกเปลี่ยนความร้อน (THERMAL PROCESS)

USSEB เราจะได้ความสัมพันธ์ระหว่าง Inlet and Outlet Temperatuves

$$q\rho_i h_i(t) - q\rho h(t) = \frac{d\left(V\rho u(t)\right)}{dt}$$

จัดรูปใหม่ให้อยู่ในรูปของอุณหภูมิ

$$q\rho_i C\rho_i T_i(t) - q\rho C\rho T(t) = \frac{d\left(V\rho C_v T(t)\right)}{dt}$$

กำหนดให้ Density and Capacity เป็นค่าคงที่ (Constant)

ซึ่งจะได้ช่วงของอุณหภูมิ (Temperature range)

$$q\rho C\rho T_i(t) - q\rho C\rho T(t) = V\rho C_v \frac{dT(t)}{dt} \qquad \text{Unknow ; T(t)} \qquad (1)$$

Function of Time จากฟังก์ชั่นจากเวลาเรากำหนดให้ Input Variable เป็น forcing function เมื่อ SSEB เราจะได้สมการ

$$q\rho C_p \overline{T}_i - q\rho C_p \overline{T} = 0 \qquad (2)$$

เมื่อแทนสมการ (2) ลงในสมการ (1) จะได้

$$q\rho C_p\left(T_i(t) - \overline{T}_i\right) - q\rho C\rho\left(T(t) - \overline{T}\right) = V\rho C_v \frac{d\left(T(t) - \overline{T}\right)}{dt} \qquad (3)$$

เราจะเห็นได้ว่า ค่าการเปลี่ยนแปลงของเทอม Deviation variable คือการเปลี่ยนแปลงของสัญญานนั่นเอง หรือเขียนใหม่ได้ว่า

(Present signal - Previous signal) = Deviation Team

(สัญญานปัจจุบัน - สัญญานในอดีต) = อัตราการเปลี่ยนแปลง

จากสมการ (3) เปลี่ยนรูปใหม่ให้อยู่ในเทอมของ Deviation Varible

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$q\rho C_p T_i(t) - q\rho C_p T(t) = V\rho C_v \frac{dT(t)}{dt} \quad (4)$$

เมื่อ $T_i(t) = T_i(t) - \bar{T}_i$

$$T(t) = T(t) - \bar{T}$$

จัดรูปสมการ (4) ใหม่

$$\frac{V\rho C_v}{q\rho C_p}\frac{dT(t)}{dt} + T(t) = T_i(t)$$

กำหนดให้

$$\tau = \frac{V\rho C_v}{q\rho C_p} \quad (5)$$

เปลี่ยนรูปให้อยู่ในเทอม S-domain โดยการใช้ลาปาส์ทรานส์ฟอร์ม

$$\tau s T(s) - \tau T(o) + T(s) = T_i(s)$$

กำหนดให้ $T(0) = 0$

$$T(s) = \frac{1}{\tau s + 1} T_i(s)$$

เราจะได้ทรายเฟอร์ฟังก์ชั่น (Transfer function)

$$\frac{T(s)}{T_i(s)} = \frac{1}{\tau s + 1} \quad \text{First-order lags} \quad (6)$$

TAKE HEAT LOSS

จากสมการ (4) เรากำหนดให้ขบวนการมีการ Loss การสูญเสียเกิดขึ้น ซึ่งเขียนสมการใหม่จะได้ว่า

$$q\rho C_p T_i(t) - Q(t) - q\rho C_p T(t) = V\rho C_v \frac{dT(t)}{dt}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซึ่ง $Q(t) = UA[T(t) - Ts(t)]$ เป็นสมการของการสูญเสีย

$$q\rho C\rho T_i(t) - UA[T(t) - T_s(t)] - q\rho C_p T(t) = V\rho C_v \frac{dT(t)}{dt}$$

เปลี่ยนรูปใหม่ให้อยู่ในเทอมของ Deviation Variable

$$q\rho C\rho T_i(t) - UA(T(t) - T_s(t)) - q\rho C_p T(t) = V\rho C_v \frac{dT(t)}{dt}$$

$$\frac{V\rho C_v}{q\rho C_p + UA}\frac{dT(t)}{dt} + T(t) = \frac{q\rho C\rho}{q\rho C_p + UA}T_i(t) + \frac{UA}{q\rho C_p + UA}T_s(t)$$

กำหนดให้

$$\tau = \frac{V\rho C_v}{q\rho C_p + UA} \quad ; \text{ Seconds}$$

$$K1 = \frac{q\rho C_p}{q\rho C_p + UA} \quad ; (^{\circ}C/^{\circ}C)$$

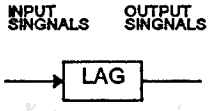
$$K2 = \frac{UA}{q\rho C_p + UA} \quad ; (^{\circ}C/^{\circ}C)$$

$$\tau\frac{dT(t)}{dt} + T(t) = K_1 T_i(t) + K_2 T_s(t)$$

$$T(s) = \frac{K_1}{\tau s + 1}T_i(s) + \frac{K_2}{\tau s + 1}T_s(s)$$

$T_{s(t)}$ จะเป็นฟังก์ชั่นของสัญญานรบกวนในระบบหากมีค่ามาก แสดงว่ามีค่าสูญเสียน้อย

FUNTCION FIRST ORDER LAG



คุณสมบัติทั่วไป Block ของ Function นี้สิ่งสำคัญที่ใช้ในการจำลองกระบวนการทางความร้อน จากการเลือกใช้ ซึ่งเขียนโดยการใช้ภาษาแอสเซมบลี้ ซึ่งหลักการทำงานจะได้อธิบายต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำอธิบาย (EXPLANATION)

Function นี้จะทำให้ Output ของ Function Block เป็น lag เปลี่ยนแปลงสัญญาน Input การทำงานอธิบายโดยสมการ

$$y = y_2 + \frac{dt(\langle s1\rangle - y_L)}{\langle s4\rangle + dt}$$

เมื่อ <S1> = แสดงค่าของ Input

<S4> = ค่าของ Time constant หน่วยเป็น Seconds

Y = แสดงค่าของ Output

YL = ค่าของ Output บน PREVIOUS CYCLE

dt = Module cycle time (Seconds)

S4 จะเป็นตัวกำหนดขอบเขตของเวลาคงที่ (Time constant) และเป็นการกำหนดเหมือน Time ซึ่งมันนำไปเป็นค่าของ Output ของ Function ซึ่งจะแผ่ออกไป= 63.2% ของค่า Input ค่าของ Output จะไม่แผ่ประมาณ 99% ของค่า Input จนกระทั่งครบ 5 Time constant ในการใช้นี้มันจะเท่ากับ 5 Time ของ S4 ก่อนที่ Output จะมีความจำเป็น แผ่ไปถึงค่า Input การคำนวณขอบเขต S4 จำเป็นสำหรับค่าของ Output ที่จะเข้าใกล้หรือเท่ากับ Input ในจำนวนของวินาที (Seconds) ที่แน่นอน ตามสมการที่จะใช้คือ

$$\langle s4\rangle = \frac{t}{5}$$

ในสมการนี้

<S4> = ขอบเขตของ Time constant สำหรับ Function LAG

t = จำนวนของวินาทีที่สำหรับ Output ที่จะแผ่ไปถึงประมาณ 99% ของค่า Input

5 = จำนวนของ Time constant ที่ต้องการสำหรับ Output ที่จะแผ่ไปถึงประมาณ 99% ของค่า Input

จากสมการ (5) ซึ่งเป็นขบวนการแลกเปลี่ยนความร้อน ซึ่งอยู่ในรูปของเวลา (Time domain)

$$\tau\frac{dT(t)}{dt} + T(t) = T_i(t) \qquad (5)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$d\mathrm{T}(t) = \frac{dt\left(T_i(t) - T(i)\right)}{\tau}$$

$d\mathrm{T}(t)$; คือค่าของการเปลี่ยนแปลงหรือ Present variable - Previous variable

τ ; คือค่าของขอบเขต Time constant ซึ่งจะมีความสัมพันธ์กับค่าของ Module cycle time (Second) ซึ่งเป็นอัตราการเปลี่ยนแปลงของ Sampling Time

จัดให้อยู่ในรูปของสัญญาน Signal

$$y - y_L = \frac{dt\left(\langle s1\rangle - y_L\right)}{s4 + \Delta T}$$

โดย Y = เอ้าท์พุทปัจจุบัน (Present output)

YL = เอ้าท์พุทอดีต (Previous output)

S1 = อินพุทปัจจุบัน (Present input)

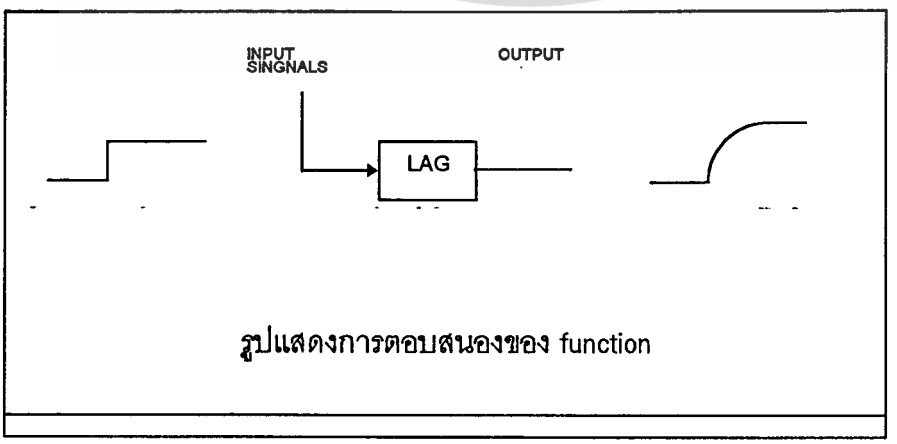
S4 = ค่าของ Time constant

dt = Module cycle time หน่วยเป็นวินาทีซึ่งเป็นเวลาของ Sampling Period Time

จัดสมการใหม่

$$y - y_L = \frac{dt\left(\langle s1\rangle - y_L\right)}{s4 + dt}$$ Function LAG

<u>การใช้งาน (APPLICATIONS)</u>



รูปแสดงการตอบสนองของ function

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปสัญญานอินพุทจะเป็นแบบ Unit step เมื่อผ่าน Function Lag จะได้สัญญานเอ้าท์พุทแบบเอ็กซ์โปแนนเชียล การคำนวณของ Function Lag นี้จะเป็นแบบ Programmed Algorithms ซึ่งรายละเอียดจะได้แสดงในหัวข้อต่อไป

วิธีการหาความสัมพันธ์ของสัญญาณ (UNITY SCALE METHOD)

ในส่วนนี้จะอธิบายถึงการคำนวณในรูปของฟังก์ชั่นบล๊อคบน Microprecessor based computing blocks และการทำ Scaling ก็มีส่วนความจำเป็นซึ่ง Scaling เป็นความสัมพันธ์ระหว่าง Input และ Output Signals

ซึ่งอาจเรียกอีกอย่างว่า "UNITY SCALE METHOD" และมีความสำคัญอย่างยิ่งในการประยุกต์ในงาน Analog Instrumentation, Pneumatin หรือ electrical และ Microprocessor-based system

เราอาจสรุปวิธีการของ Unity scale method เป็นขั้นตอนได้ดังนี้

1. เขียนสมการที่ต้องการแก้และช่วง (Rang) จากตัวแปรของกระบวนการกำหนดชื่อของสัญญานจากตัวแปรของกระบวนการ
2. หาความสัมพันธ์ของตัวแปรกระบวนการ โดยการใช้การลดรูปสมการ (Normalized equation)
3. แทนค่าหรือปรับแต่งสมการ และแทนค่าเพื่อหาสัญญาน Output

- Lincarized tank model equations in terms of deviation variables

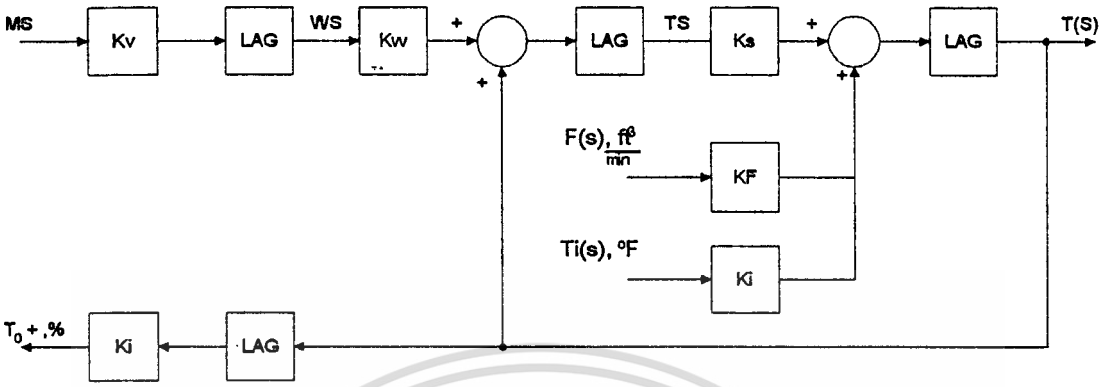
$$\rho C_p \frac{dT(t)}{dt} = \bar{f}\rho C_p T_i(t) + \rho C_p\left(\overline{T}_i - \overline{T}\right)F(t) + UATs(t) - \left(UA + \bar{f}\rho C_p\right)T(t)$$

$$C_M \frac{dTs(t)}{dt} = \lambda W(t) - UATs(t) + UAT(t)$$

- lincarization of the valve equation

$$(t) = \overline{W}_{MAX}(\ln\alpha)\alpha^{\overline{VP}-1\cdot VP(t)}$$
$$= \overline{W}(\ln\alpha)VP(t)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Block diagrame

จากตัวแปรกระบวนการ เราจะกำหนด Ranges จากตัวแปรดังนี้

Signal	Variable	Rang	Design Conditions
S1	MS	0-100%	–
S2	WS	0-165.2 lbm/min	42.2 lb/min
S3	WS1	0-165.2 lbm/min	42.2 lb/min
S4	WS2	0-165.2 lbm/min	42.2 lb/min
S5	TS	0-500 °F	230 °F
S6	TS1	0-500 °F	230 °F
S7	TS2	0-500 °F	230 F150 °F
S8	T0	100-200 °F	150 °F
S9	TO1	100-200 °F	150 °F
S10	TO2	100-200 °F	150 °F
S11	PV	100-200 °F	150 °F
S12	PV1	100-200 °F	150 °F
S13	PV2	100-200 °F	150 °F
S14	Fd	15-25 ft^3/min	15 ft^3/min
S15	Ti	100-200 °F	100 °F
S16	TD	—	—

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ความสัมพันธ์ระหว่างตัวแปรกระบวนการกับสัญญาน ซึ่งตัวแปรจะมีการเปลี่ยนแปลงระหว่างค่าสูงและค่าต่ำจาก Range ซึ่งสามารถเขียนเป็นสมการได้

$$\text{Signal} = \frac{\text{Process variable - Low value of range}}{\text{Span}}$$

เราจะได้สัญญานดังนี้

$S_1 = \frac{MS}{100}$ or $MS = 100S_1$

$S_2 = \frac{WS}{165.2}$ or $WS = 165.2S_2$

$S_3 = \frac{WS1}{165.2}$ or $WS1 = 165.2S_3$

$S_4 = \frac{WS2}{165.2}$ or $WS2 = 165.2S_4$

$S_5 = \frac{TS}{500}$ or $TS = 500S_5$

$S_6 = \frac{TS1}{500}$ or $TS1 = 500S_6$

$S_7 = \frac{TS2}{500}$ or $TS2 = 500S_7$

$S_8 = \frac{TO-100}{100}$ or $TO = 100S_8+100$

$S_9 = \frac{TO1-100}{100}$ or $TO1 = 100S_9+100$

$S_{10} = \frac{TO2-100}{100}$ or $TO2 = 100S_{10}+100$

$S_{11} = \frac{PV-100}{100}$ or $PV = 100S_{11}+100$

$S_{12} = \frac{PV1-100}{100}$ or $PV1 = 100S_{12}+100$

$S_{13} = \frac{PV2-100}{100}$ or $PV2 = 100S_{13}+100$

$S_{14} = \frac{Fd-15}{10}$ or $Fd = 10S_{14}+15$

$S_{15} = \frac{Ti-100}{100}$ or $Ti = 100S_{15}+100$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จาก Block Diagrame เราจะได้สมการดังนี้

$$WS = Kv \times MS$$

$$WS1 = WS2 + \frac{DT(WS - WS2)}{S4 + DT}$$

$$TS = TO2 + (WS1 \times Kw)$$

$$TS1 = TS2 + \frac{DT(TS - TS2)}{S5 + DT}$$

$$TD = Fd \times KF + Ti \times Ki$$

$$TO = (TS \times KS) + TD$$

$$TO1 = TO2 + \frac{DT(TO - TOS)}{S6 + DT}$$

$$PV1 = PV2 + \frac{DT(PV - PV2)}{S7 + DT}$$

จากสมการ (1)

$WS = k_V \times MS$ (1)

กำหนดให้ k_V = 1.652

MS = 100 S1

WS = 165.2 S2

เมื่อแทนค่าในสมการ (1) จะได้สัญญาน Signal

165.2 S2 = 165.2 (100 S1)

S2 = 100 S1

จากสมการ (2)

เป็น Block ของ First order Lage

$$S1 = WS2 + \frac{DT(WS - WS2)}{S4 + DT}$$

กำหนดให้ S4 = t/5 ; *t* คือ Time constant

$\tau_v = 0.20\ min$

∴ S4 = 2.4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

WS1 = 165.2 S3

WS2 = 165.2 S4

แทนค่าในสมการ (2) เปลี่ยนรูปให้อยู่ในรูปของ Signal

WS1 = WS2 + 0.2941 (WS-WS2)

165.2S3 = 0.2941 (165.2S2) + 0.7059 (165.2S4)

S3 = 0.2941S2 + 0.7059S4

จากสมการ (3)

*TS = TO2 + (WS1*KW)*

กำหนดให้

TS = 500S5

TO2 = 100S10100

WS1 = 165.2S3

KW = 1.905

แทนค่าในสมการ (3) เปลี่ยนรูปให้อยู่ในรูปของ Signal

500S5 = (100S10$_0$ + 100) + 1.905(165.2S3)

S5 = 0.6294S3 + 0.2S10+.02

จากสมการ (4)

$$TS1 = TS2 + \frac{DT(TS - TS2)}{S5 + DT}$$

กำหนดให้ S5 = 6.288

DT = 1

TS = 500S5

TS1 = 500S6

TS2 = 500S7

แทนค่าในสมการ (4) เพื่อเปลี่ยนรูปให้อยู่ในรูปของ Signal

TS1 = TS2 + 0.1372 (TS-TS2)

TS1 = 0.8628T S2 + 0.1372 TS

S6 = 0.8628 S7 + 0.1372 S5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากสมการ (5)

$$TD = Fd*K_F + Ti*K_i$$

กำหนดให้

Fd = 10 S14+15

Ti = 100 S15+100

K_i = 0.617

K_F = -2.06

แทนค่าสมการ (5)

$$TD = 61.7\ S15_6 - 20.6\ S14 + 3\ 0.8 \qquad (5.1)$$

จากสมการ (6)

T0 = (TS1*KS)+TD

กำหนดให้ T0 = 100 S8 + 100

TS1 = 500 S6

KS = 0.383

TD = 61.7 S15 - 20.6 S14 +30.8

แทนค่าในสมการ (6)

S8 = 1916 S6 + 0.617 S15 - [0.206 S14 + 0.692]

จากสมการ (7)

$$T01 = T02 + \frac{DT(T0+T02)}{S6+DT}$$

กำหนดให้ S6 = 59.16

DT = 1

T0 = 100 S8 + 100

T01 = 100 S9 + 100

T02 = 100 S10+100

แทนค่าในสมการ (7)

T01 = T02+0.0166(T0-T02)

T01 = 0.9834 T02+0.0166T0

100S9+100 = 0.9834(100S10+100)+0.0166(100S8+100)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

S9 = 0.9834 S10 + 0.0166 S8

จากสมการ (8)

$$PV1 = PV2 + \frac{DT(PV - PV2)}{S7 + DT}$$

กำหนดให้ PV = 100 S11 + 100

PV1 = 100 S12 + 100

PV2 = 100 S13 + 100

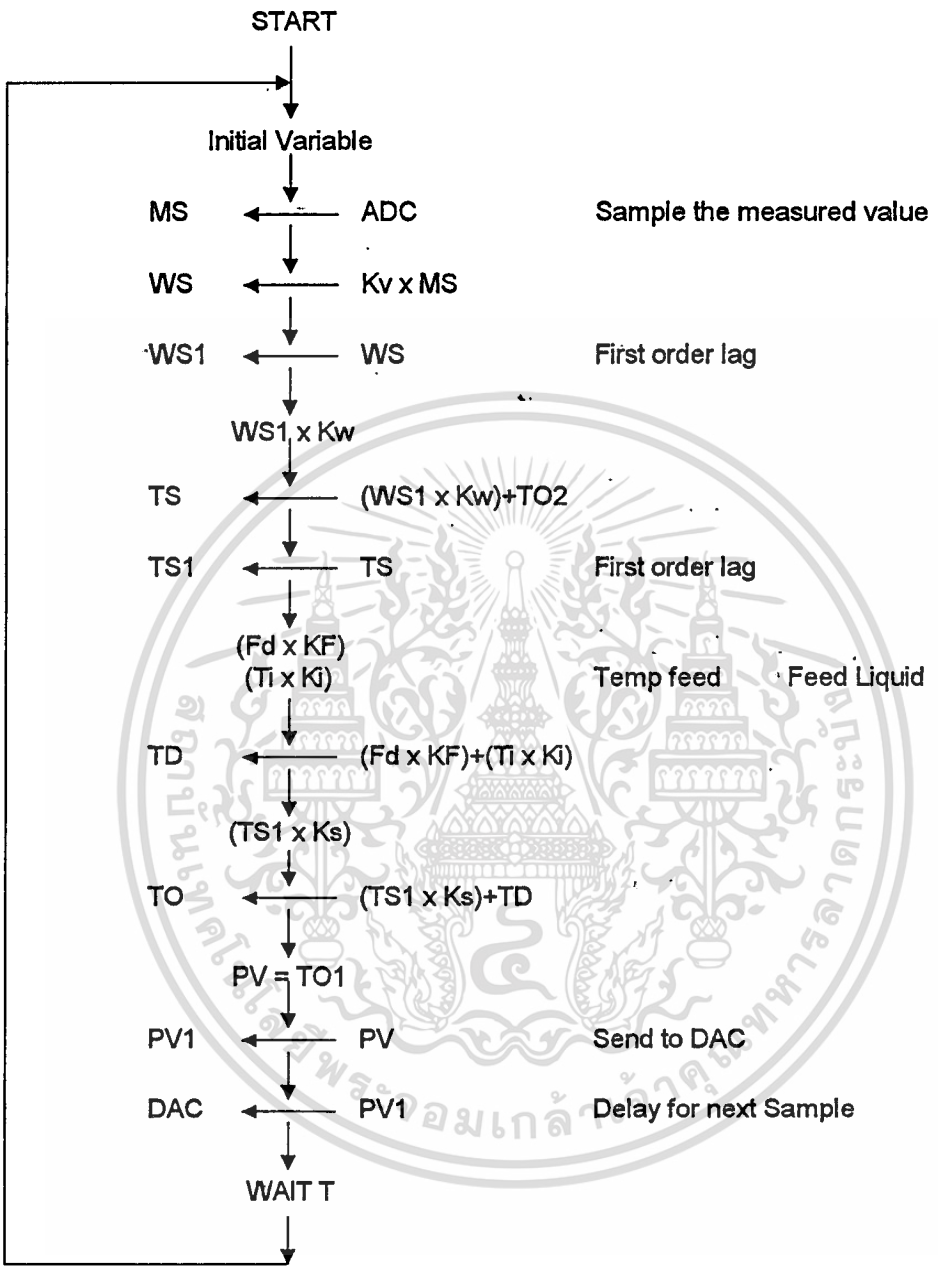
DT = 1

แทนค่าในสมการ (8) จะได้สัญญาน Signal

S12 = 0.9 S13 + 0.1 S11

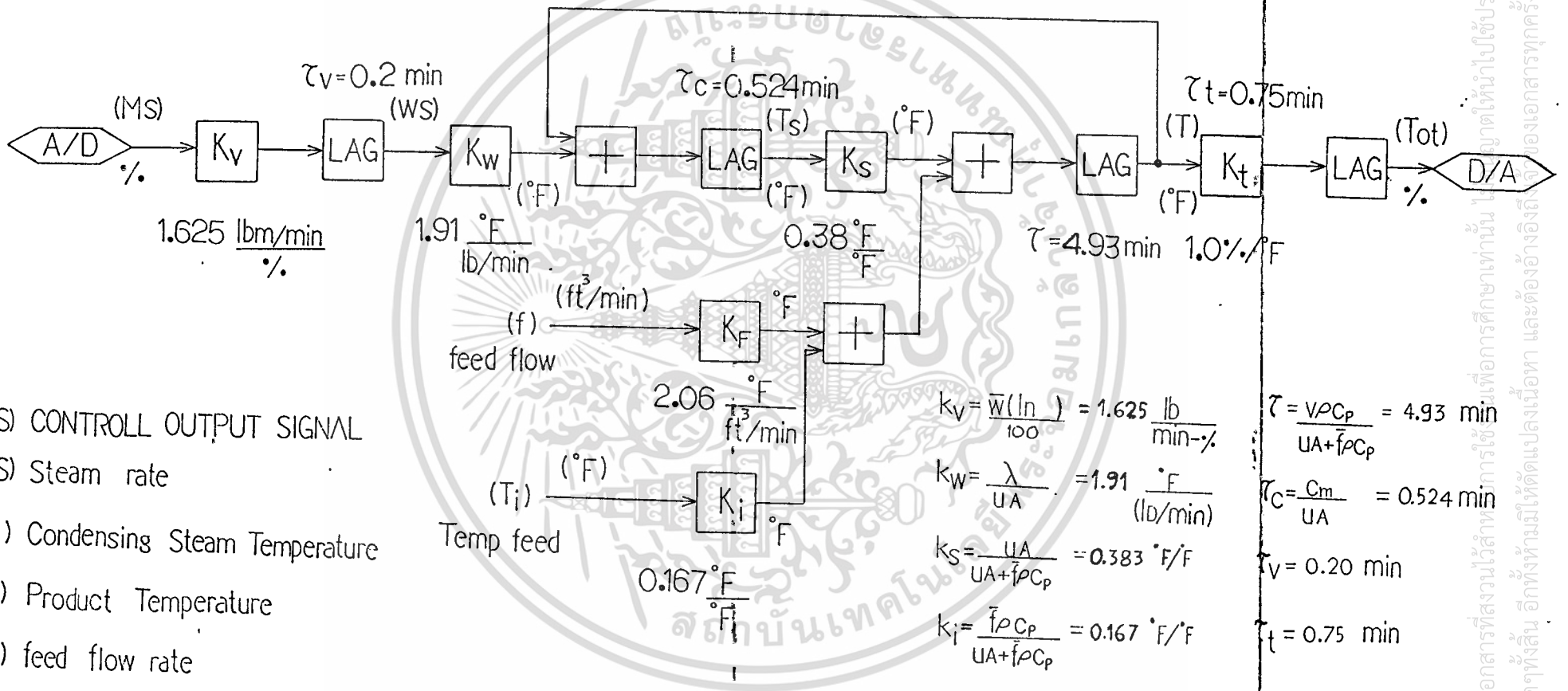
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สรุป การทำงานของกระบวนการที่เลือกใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Model Heat Exchange



(MS) CONTROLL OUTPUT SIGNAL

(WS) Steam rate

(Ts) Condensing Steam Temperature

(T) Product Temperature

(f) feed flow rate

(Ti) Temp feed Liquid

(Tot) Transmitter output signal

$k_V = \frac{\bar{W}(\text{In})}{100} = 1.625 \frac{\text{lb}}{\text{min-\%}}$

$k_W = \frac{\lambda}{UA} = 1.91 \frac{^\circ F}{(\text{lb/min})}$

$k_S = \frac{UA}{UA+\bar{f}\rho C_p} = 0.383\ ^\circ F/^\circ F$

$k_i = \frac{\bar{f}\rho C_p}{UA+\bar{f}\rho C_p} = 0.167\ ^\circ F/^\circ F$

$k_F = \frac{\rho C_p(\bar{T}i-\bar{T})}{UA+\bar{f}\rho C_p} = -2.06 \frac{^\circ F}{ft^3/\text{min}}$

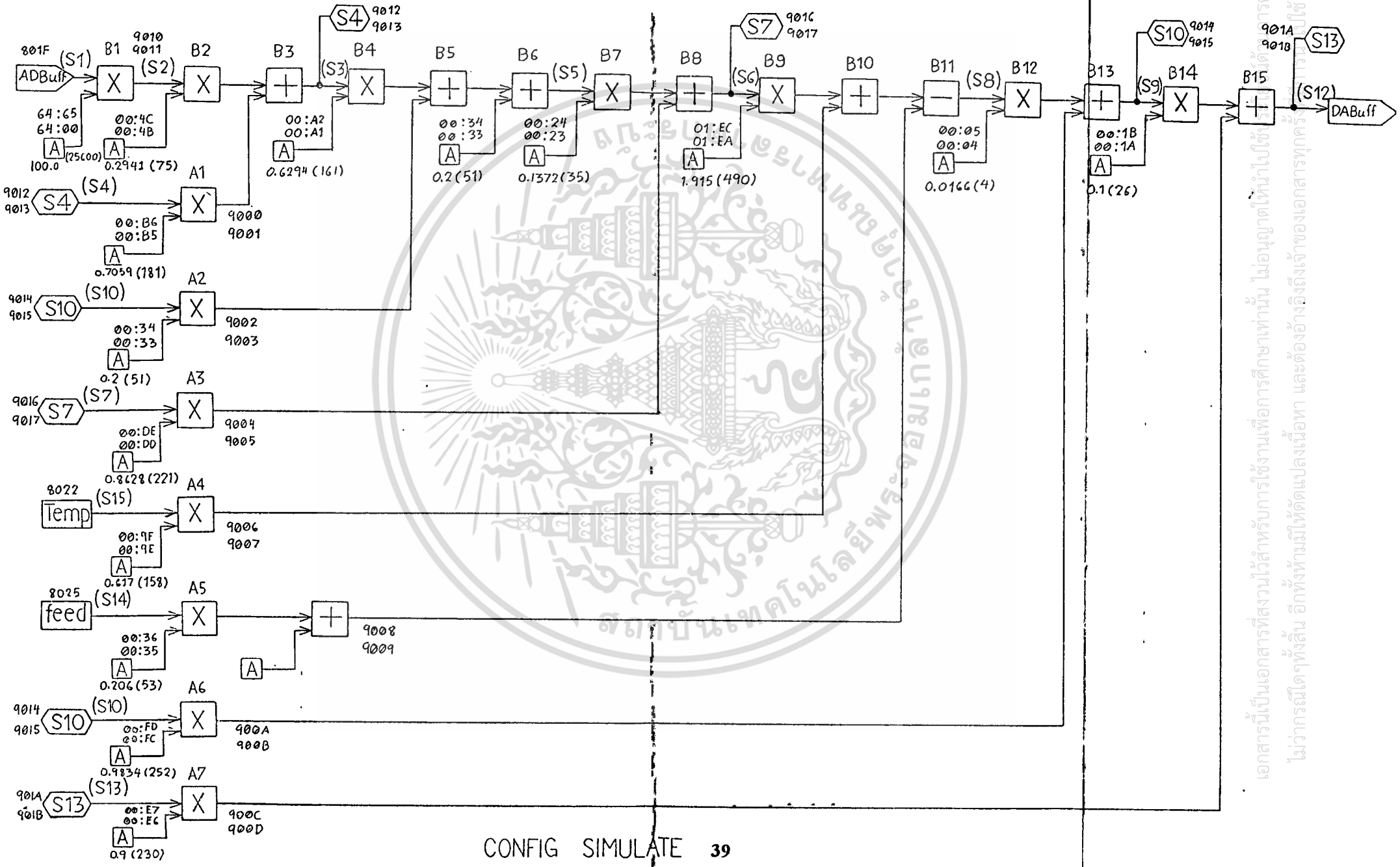
$k_t = \frac{100-0}{200-100} = 1.0\ \%/^\circ F$

$\tau = \frac{V\rho C_p}{UA+\bar{f}\rho C_p} = 4.93$ min

$\tau_C = \frac{Cm}{UA} = 0.524$ min

$\tau_V = 0.20$ min

$\tau_t = 0.75$ min


801F
ADBuff
(S1)
B1
9010
9011
(S2)
B2
B3
S4
9012
9013
(S3)
B4
B5
B6
(S5)
B7
B8
S7
9016
9017
(S6)
B9
B10
B11
(S8)
B12
B13
S10
9014
9015
(S9)
B14
B15
9018
S13
(S12)
DABuff
64:65
64:00
100.0
(25600)
00:4C
00:4B
0.2941 (75)
00:A2
00:A1
0.6294 (161)
00:34
00:33
0.2 (51)
00:24
00:23
0.1372(35)
01:EC
01:EA
1.915 (490)
00:05
00:04
0.0166 (4)
00:1B
00:1A
0.1 (26)
A1
9012
9013
S4
(S4)
00:B6
00:B5
0.7059 (181)
9000
9001
A2
9014
9015
S10
(S10)
00:34
00:33
0.2 (51)
9002
9003
A3
9016
9017
S7
(S7)
00:DE
00:DD
0.8628 (221)
9004
9005
A4
8022
Temp
(S15)
00:9F
00:9E
0.617 (158)
9006
9007
A5
8025
feed
(S14)
00:36
00:35
0.206 (53)
9008
9009
A6
9014
9015
S10
(S10)
00:FD
00:FC
0.9834 (252)
900A
900B
A7
901A
901B
S13
(S13)
00:E7
00:E6
0.9 (230)
900C
900D

บทที่ 3
การออกแบบตัวจำลองกระบวนการทางความร้อน โดยใช้ไมโครคอนโทรลเลอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

การออกแบบตัวจำลองกระบวนการ โดยการใช้ไมโครคอนโทรลเลอร์ (MCS - 51)

— การออกแบบซอฟแวร์ของตัวจำลองกระบวนการ

— ไมโครคอนโทรลเลอร์และซอฟแวร์

การออกแบบเราใช้ไมโครโปรเซสเซอร์ 8032 ซึ่งเป็นไมโครโปรเซสเซอร์ 8 บิต โดยอาศัยชุดพัฒนาของ ET-8032 เป็นต้นแบบ สาเหตุที่เลือกใช้ ET-8032 บอร์ดนั้น ได้แก่

1. ง่ายต่อการศึกษาการทำงานของ CPU
   — ทั้งหน่วยความจำภายในและภายนอก CPU ผู้ใช้สามารถเข้าถึงได้ด้วยความรวดเร็วไม่ซับซ้อน
   — ทั้งยังสามารถศึกษาการจัดหน่วยความจำแบบรวมเพจ (DATA MEMORY และ PROGRAM MEMORY อยู่หน้าเดียวกัน 64K) หรือจะเป็นแบบแยกเพจคือ PROGRAM MEMORY และ DATE MEMORY แยกกันสามารถมองหน่วยความจำได้สูงสุด 128K

2. ความอ่อนตัวของ HARDWARE
   — ตำแหน่งของไอซีหน่วยความจำสามารถเลือกได้ว่าจะเป็น PROGRAM MEMORY หรือ DATA MEMORY

3. REMOTE COMMAND
   — คอมพิวเตอร์ส่วนบุคคล ยังสามารถศึกษาการทำงานของ CPU ในตระกูล MCS-51 นี้ผ่านทางเครื่องคอมพิวเตอร์ส่วนบุคคล (PC) โดยผ่าน FUNTION REMOTE ของ ET-8032 ทำให้สะดวกมากขึ้น

การเขียนโปรแกรมและภาษาสำหรับ ET-8032

ภาษาเครื่อง เป็นภาษาขั้นต่ำสุดที่ผู้ใช้สามารถสั่งงานให้คอมพิวเตอร์ทำงานตามความต้องการ ถึงแม้จะยุ่งยาก แต่ก็เป็นภาษาเดียวที่ CPU ทำงานได้เร็วที่สุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาษาแอสเซมบลี้ มนุษย์หลีกเลี่ยงความยุ่งยากสับสนในภาษาเครื่องโดยกำหนดรหัสหรือคำสั่งที่ใช้แทนภาษาเครื่องโดยให้มีความหมายใกล้เคียงกับภาษาของมนุษย์มากที่สุดเรียกว่าภาษาแอสเซมบลี้ และสร้างตัวแปลภาษาแอสเซมบลี้ เป็นภาษาเครื่องอีกทีเรียกว่า "แอสเซมเบลอร์"

รูปแบบของภาษาแอสเซมบลี้

ใน 1 บรรทัดของภาษาแอสเซมบลี้ ประกอบด้วย

ลาเบล นีโมนิค โอเปอแรนด์ 1 โอเปอแรนด์ 2 คำอธิบาย

ลาเบล : จะมีก็ได้ไม่มีก็ได้เป็นจุดอ้างถึงแอดเดรส

นีโมนิค : รหัสคำสั่งหรือเรียกว่า OPERATOR

โอเปอแรนด์ 1และ 2 : คือตัวถูกกระทำ บางคำสั่งอาจจะมี โอเปอแรนด์ 1 เพียงตัวเดียว

ตัวอย่าง ภาษาแอสเซมบลี้

```
        ORG 8oooH
ST1 :   MOV   DPTR,#0E060H   ;   POINT TO E060H
        MOV   A,#12H         ;   DATA
ST2 :   MOVX  @DPTR,A
        INC A
        DJNZ  ST2
        END
```

ภาษาแอสเซมบลี้ อาจถูกเขียนด้วย TEXT EDITOR ตัวใดก็ได้ เพราะภาษาแอสเซมบลี้ ต้องการเพียงรูปแบบของภาษาที่เป็นข้อความ เพื่อให้ตัวแปลทำการแปลอีกที เมื่อนำโปรแกรมที่เขียนไปผ่านตัวแปล (แอสเซมเบลอร์) จะได้ไฟล์อีก 1 หรือ 2 ชนิด ขึ้นอยู่กับตัว แอสเซมเบลอร์นั้น ๆ PROJECT นี้ เราเลือกใช้ SXA51 เป็นตัวแอสเซมเบลอร์

การจัดการหน่วยความจำ

สถาปัตยกรรมของ MCS-51 ได้แบ่งหน่วยความจำมาให้บนชิป พร้อมทั้งสามารถที่จะขยายหน่วยความจำภายนอกได้ โดยทั่วไปการกำหนดแลขที่อยู่ของเบอร์นี้ จะมีคำสั่งเพื่อช่วยให้ทำงานได้รวดเร็ว และสามารถใช้คำสั่งได้อย่างเต็มประสิทธิภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MCS-51 แบ่งตามพื้นฐานหน่วยความจำของการกำหนดเลขที่อยู่แอดเดรสได้เป็น 3 ส่วนที่ประกอบด้วยเนื้อที่

64 กิโลไบต์หน่วยความจำโปรแกรม

64 กิโลไบต์หน่วยความจำข้อมูลภายนอก

256 ไบต์ เป็นหน่วยความจำข้อมูลภายในส่วนเบอร์ 8032 มีขนาด 384 ไบต์

MEMORY MAP

0000H	PROGRAM MEMORY
7FFFH	
8000H	DATA-LCD
9EFFH	
9F00H	Sub-LCD
9FFFH	
A000H	Sub-WORKING
DFFFH	
E000H	PORT
FFFFH	

I/O MAP

E000H - E010H = PORT A

E001H - E011H = PORT B

E002H - E012H = PORT C

E003H - E013H = CONTROL PORT

E020H - E030H = PORT A

E021H - E031H = PORT B

E022H - E032H = PORT C

E023H - E033H = CONTROL PORT

E080H - E090H = LCD

E0C0H - E0FFH = RTC (DS1287)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การใช้งานพื้นที่แรมภายใน (Interntal date RAM)

การใช้งานพื้นที่แรมภายใน (Interntal date RAM) ของไมโครคอนโทรลเลอร์ตระกูล MCS-51 จัดแบ่งหน่วยความจำภายในของไมโครคอนโทรลเลอร์แสดงเป็นตารางได้ดังนี้

หน่วยความจำแรมภายใน (Interntal RAM) สามารถนำมาใช้ประโยชน์โดยมาใช้เป็นพื้นที่กำหนดตัวแปรต่าง ๆ ซึ่งใช้เป็นตัวจำลองกระบวนการ

ADDRESS RAM แอดเดรสแรม	VARIABLE ตัวแปร	REMARK คำบรรยาย
10H	KEYBuff	ตัวแปรของ Key Borad
11H	DABuff	ตัวแปรของ เอ้าพุท
12H	ADBuff	ตัวแปรของ อินพุท
13H	TO	ตัวแปรของอุณหภูมิของไหล
14H	TS	ตัวแปรของอุณหภูมิไอน้ำ
15H	WS	ตัวแปรของอัตราการไหล
16H	FEED	ตัวแปรของ FEED Liguid
17H	TEMP	ตัวแปรของ Temp Liguid

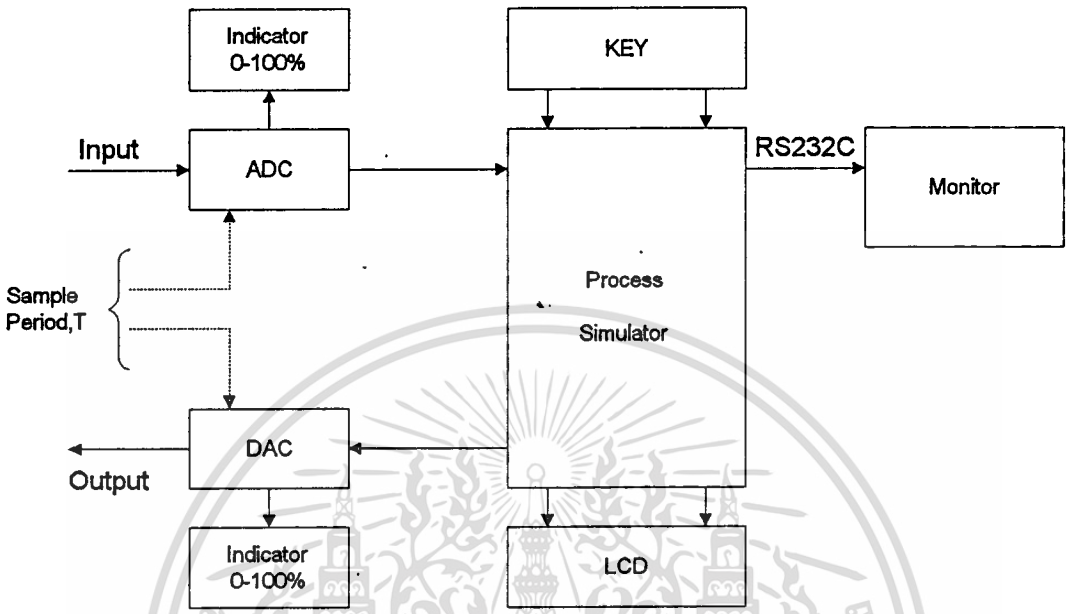
ตารางแสดง Internal RAM

แอสเดรส	หมายเลขบิต	ตัวแปร	คำบรรยาย
20H	0	AD-DWFlag	แฟลกกำหนด Analog
	1	AD-UPFlag	แฟลกกำหนด Analog
		DW-Temp	แฟลกกำหนดการลดของอุณหภูมิ
	3	UP-Temp	แฟลกกำหนดการเพิ่มของอุณหภูมิ
	4	DW-Feed	แฟลกกำหนดการลดของอัตราการไหล
	5	UP-Feed	แฟลกกำหนดการเพิ่มของอัตราการไหล
	6	MV-Flag	แฟลกกำหนด Auto/Manual
	7	-	

ตารางกำหนดบิตแฟลก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โครงสร้างของตัวจำลองกระบวนการ



รูป บล๊อคไดอะแกรมของตัวจำลองกระบวนการ

การออกแบบตัวจำลองกระบวนการจะใช้ไมโครโปรเซสเตอร์ตระกูล MCS-51ของบริษัท อินเทล โดยใช้สัญญานนาฬิกา 12 เมกกะเฮิรตซ์ จึงมีความเร็วพอเพียงที่จะเขียนซอฟแวร์โปรแกรม โปรแกรมนี้ทำงานในลักษณะเป็นเวลาจริง (Real time) โดยอาศัย SHIP DS1287 เป็นตัวจัดการ Period Time ซึ่งในกระบวนการประมวลผลจะอาศัยเป็นตัว Sampling ในแต่ละครั้งของการประมวลผล

การป้อน Input สามารถกระทำได้ 2 Mode คือ 1) Auto 2) Manual โดยจะมี LED แสดงสถานะของ Mode การทำงาน ถ้าหากอยู่ใน Manual Mode โปรแกรมจะอนุญาตให้มีการป้อน Key Board ของสัญญาน MV ได้ ส่วนสัญญานอื่น ๆ ดูในรายละเอียดเรื่อง Key Board

การจัดการทำงานของโปรแกรม

จากบล๊อคไดอะแกรม เราจะแบ่งออกเป็นส่วน ๆ เพื่อที่จะได้อธิบายหน้าที่ของการทำงาน

1. ส่วนของ Input
2. ส่วนของ Output
3. ตัว Process Simulator

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. Key Borad

5. LCD

วงจรแสดงผล (Display circuit)

LCD (LIQUID CRYSTAL DISPLAY)

ปัจจุบัน LCD ได้ถูกนำมาใช้ในการแสดงผลมากขึ้น เนื่องจากราคาถูกลง และใช้งานง่าย ซึ่งเราอาจจะพบเห็นได้ทั่วไปในรูปของ นาฬิกา เครื่องคิดเลข เกมส์กด จนถึงเครื่องคอมพิวเตอร์

ข้อดีของ LCD

1. บางเบา พกพาได้สะดวก

2. ใช้พลังงานน้อย เช่น อาจจะใช้ VOLTAGE ต่ำขนาด 2 หรือ 3V และกระแสเพียงไม่กี่มิลลิแอมป์ก็สามารถใช้งานได้

3. ราคาถูก

4. ใช้งานในช่วงอุณหภูมิที่กว้างได้

5. มีคำสั่งพิเศษสำหรับอำนวยความสะดวกมากมายเช่น clear display, home cursor on off cursor, blink character และอื่น ๆ อีก

6. สามารถแสดงผลเป็นตัวอักษรภาษาอังกฤษและตัวเลขได้ 160 ตัวและสัญญลักษณ์พิเศษอีก 32 ตัว รวมทั้งสามารถกำหนดอักษรที่ออกแบบได้เองอีก 8 ตัว

การต่อเข้ากับระบบไมโคร

LCD MODULE จะต่อเข้ากับระบบไมโครได้ 2 ลักษณะ คือแบบ MEMORY MAP โดยผ่าน LCD BUS ขนาด 20 PIN และแบบ I/O PROT โดยผ่าน 8255 ขนาด 26 PIN ทั้งสองนี้จะมีข้อดีและข้อเสียแตกต่างกันไป โดยแต่ละแบบจะมีหลักการดังนี้

การต่อแบบ MEMORY MAP

1. สามารถต่อเข้ากับ CHIP เบอร์ทั่ว ๆ ไปได้ เช่น 8051 หรือ z80 โดยจะทำให้ระบบไมโครมองเห็น LCD MODULE ในลักษณะของ MEMORY ได้ทันที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. ผู้ใช้สามารถเขียนและอ่านข้อมูลจาก LCD MODULE ได้ทำให้มองเสมือนว่าเป็น MEMORY BUFFER ไปในตัว

3. เนื่องจากสามารถอ่านข้อมูลกลับได้ จึงทำให้สามารถตรวจสอบ FLAG ความพร้อมในขณะที่ LCD MODULE กำลังทำงานได้

4. ใช้ได้กับบอร์ดที่มี LCD BUS มาให้พร้อมเท่านั้น

5. ทำให้กินพื้นที่ของหน่วยความจำไปส่วนหนึ่ง และต้องมีการ DECODE ละเอียดพอควร

6. การจัดขาสัญญานจะต้องเป็นไปตามแบบ CHIP แต่ละเบอร์ด้วย

<u>การต่อแบบ I/O PORT</u>

1. สามารถต่อเข้ากับ I/O PORT ใด ๆ ก็ได้ โดยใช้สัญญานจำนวน 11 เส้น และใช้โปรแกรมเป็นตัวสร้างสัญญานขึ้นมาให้ตรงกับข้อกำหนดของ LCD MODULE

2. ผู้ใช้จะเขียนข้อมูลให้ LCD MODULE ได้อย่างเดียว ซึ่งผู้ใช้ควรจะกำหนด MEMORY ส่วนหนึ่งให้เป็นเสมือน BUFFER ให้กับ LCD MODULE อีกที

3. เนื่องจากไม่สามารถอ่านข้อมูลกลับได้ จึงต้องใช้การหน่วงเวลาของระบบไมโครเอง เพื่อรอให้ LCD MODULE กระทำขบวนการต่าง ๆ

4. ใช้ได้กับบอร์ดทั่ว ๆ ไปที่มี PORT

5. ไม่เปลืองส่วนของ MEMORY ในการใช้งาน

6. การจัดขาสัญญานกระทำได้อย่างอิสระ

<u>ความเข้าใจพื้นฐาน</u>

1. การเขียนข้อมูลให้กับ LCD MODULE จะแบ่ง 2 ลักษณะคือ INSTRUCTION และ DATA โดยจะกำหนดด้วยขาสัญญาน RS คือถ้า RS = 0 จะหมายถึงส่งสัญญานควบคุม (INSTRUCTION) หรืออ่านค่า FLAG สภาพการทำงานของ LCD MODULE และถ้า RS = 1 จะหมายถึงการเขียน หรืออ่าน DATA กับ LCD MODULE

2. หลักการในการเขียนข้อมูลให้ LCD MODULE นี้ คือเมื่อมีการเขียนข้อมูลไปแล้ว ตัว LCD MODULE จะต้องใช้เวลาในการทำงานชั่วขณะหนึ่ง (จามค่า EXECUTE TIME ในตาราง) ซึ่งระบบไมโครสามารถตรวจสอบได้จาก BUSY FLAG (BF) และถ้าเรียบร้อยแล้ว จึงจะสามารถเขียนข้อมูลอันต่อไปได้ ในกรณีที่การต่อวงจรเป็นแบบ I/O PORT คือไม่สามารถอ่านข้อมูลย้อนกลับได้ ระบบไมโครก็จะต้องใช้วิธีการหน่วงเวลาแทน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. การเขียนข้อมูลให้กับ LCD MODULE นี้ สามารถทำได้ทั้งแบบ 8 BIT และ 4 BIT โดยกรณี 4 BIT จะใช้สายสัญญาน DATA เพียง 4 เส้น คือ DB4-DB7 (ใช้สำหรับระบบไมโครแบบ 4 BIT หรือการประหยัดสาย).การเขียนข้อมูลจะกระทำเหมือนกับ 8 BIT เพียงแต่ให้เขียน 2 ครั้งคือ DB4-DB7 ก่อน แล้วตามด้วย DB0-DB3 และจะต้องกำหนดคุณสมบัติตามค่า DL ในคำสั่ง FUNCTION SET ด้วย

4. DDRAM (DISPLAY DATA RAM) คือหน่วยความจำภายในตัว LCD MODULE ที่เป็น BUFFER ของข้อมูล โดยถ้าเขียนรหัส ASCII ใด ๆ ลงไปในหน่วยความจำนี้ ก็จะปรากฏเป็นตัวอักษรที่แผงแสดงทันที

5. CGRAM (CHARACTER GENERATOR RAM) คือหน่วยความจำภายในตัว LCD MODULE สำหรับเก็บภาพตัวอักษาที่ผู้ใช้สามารถสร้างได้เอง (8 ตัว) โดยจะอ้าง ADDRESS ได้ทั้งหมด 64 BYTE คือตัวอักษร คูณกับ 8 ROW

วงจร Analog to Digital (ADC)

ทฤษฎี

ลักษณะโดยทั่ว ๆ ไป ของ D/A และ A/D จะมีความเกี่ยวข้องกันอย่างมากเมื่อเราสามารถนำสัญญาน DIGITAL เปลี่ยนเป็น ANALOG ได้แล้ว ในทางกลับกัน ก็ควรจะเปลี่ยนสัญญาน ANALOG เป็น DIGITAL ได้ด้วย ซึ่งในจุดนี้จะทำให้เราทำการอ่านค่าข้อมูลทาง ANALOG ได้ เช่นค่าความดัน, ค่าอุณหภูมิ โดยจะมีประโยชน์อย่างมากต่องานทางด้านวิศวกรรม

คุณสมบัติของ A/D จะมีลักษณะเหมือนกับ D/A และจะมีคุณสมบัติพิเศษอีก 1 อย่าง คือ CONVERSION TIME ดังนี้หมายถึงช่วงเวลาที่ A/D ใช้ในการแปลงสัญญาน ANALOG เป็น DIGITAL ได้ 1 ค่า A/D ที่ดีควรมี CONVERSION TIME น้อยที่สุดเท่าที่จะน้อยได้

DIGITAL TO ANALOG

ทฤษฎี

ไมโครโปรเซสเซอร์จะมีการทำงานในเชิงของ DIGITAL เสอ ซึ่งมีระดับสัญญานเป็น 0 และ 1 แต่ในการประยุกต์ใช้งานจริง มักจะต้องเกี่ยวข้องกับสัญญานที่เป็น ANALOG ซึ่งสัญญานที่มีความแตกต่างหลายระดับ สัญญานเหล่านี้ได้แก่ อุณหภูมิ, ความดัน, น้ำหนักเสียง ในทางอิเลคทรอนิคส์เราสามารถเปลี่ยน VOLTAGE ให้อยู่ในรูปของสัญญานเหล่านี้ได้ และระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คอมพิวเตอร์ก็ต้องเปลี่ยนสัญญาน 0 และ 1 ให้เป็นสัญญาน VOLTAGE ได้เช่นกัน ซึ่งสัญญาน 0 และ 1 นี้ ไมโครจะอยู่ในรูปของข้อมูลทางคอมพิวเตอร์

<u>คุณสมบัติทั่ว ๆ ไปของ D/A จะกล่าวถึงเป็นข้อ ๆ ดังนี้</u>

1. RESOLUTION คือ ความสามารถในการแบ่งแยกระดับของสัญญานหรืออีกนัยหนึ่งก็คือจำนวนบิทของสัญญาน DIGITAL นั่นเอง เช่น ขนาด 8 บิต จะหมายความว่า สามารถแยกสัญญาน ANALOG ไว้เป็น 256 ระดับ (2^8) ถ้าขนาด 12 บิต ก็จะแยกสัญญานได้เป็น 4096 ตามระดับ (2^{12})ค่า RESOLUTION บางครั้งก็จะแสดงอยู่ในรูปของเปอร์เซนต์ โดยขนาด 8 บิต จะมีค่าเท่ากับ 100/256 คือ 0.39%

2. FULL SCALE OUTPUT VOLTGE คือ ค่าแรงดันสูงสุดที่จะเป็นไปได้ของสัญญาน ANALOG สมมติว่าใช้ D/A ขนาด 8 บิต เราจะเขียนเป็นสูตรได้ดังนี้

$$VO = Vref\left(\frac{A1}{2} + \frac{A2}{4} + \frac{A3}{8} + \frac{A4}{16} + \frac{A5}{32} + \frac{A6}{64} + \frac{A7}{128} + \frac{A8}{258}\right)$$

A1 ถึง A9 คือสัญญาน DIGITAL มีค่าเป็น 0 หรือ 1 ตามข้อมูลในที่นี้ถ้าสมมติ Vref = 10V ค่าสูงสุดที่เป็นไปได้คือ ให้A1 ถึง A8 = 1 ดังนี้

$$VO = 10 \times \frac{256}{258} = 9.9609\ V$$

นั่นหมายความว่า สัญญาน ANALOG จะมีค่าสูงสุดน้อยกว่า Vref อยู่ 1 ระดับเสียง และในที่นี้ความแตกต่างของแต่ละระดับจะเท่ากัน

$$Vdiff = \frac{10}{} = 0.03906\ Volt$$

3. ACCURACY คือ ค่าเปรียบเทียบระหว่าง VOLT จริง ๆ ที่ได้กับ Volt ที่กำหนดให้เป็น ถ้า D/A มีคุณสมบัติเป็น 10V และ +0.15% ACCURACY นั่นหมายถึง D/A ชุดนี้จะมีโอกาสผิดพลาดได้สูงคือ 0.0015*10 = 0.015V ปกติค่า ACCURACY ในทางจุดปกติแล้ว ไม่ควรจะมีค่ามากกว่าครึ่งหนึ่งของระดับสัญญาน ANALOG หรือเท่ากับ +1/2 ของบิตที่ต่ำที่สุด (LBS) ในที่นี้คือ 0.039/2 = 0.0195 V

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

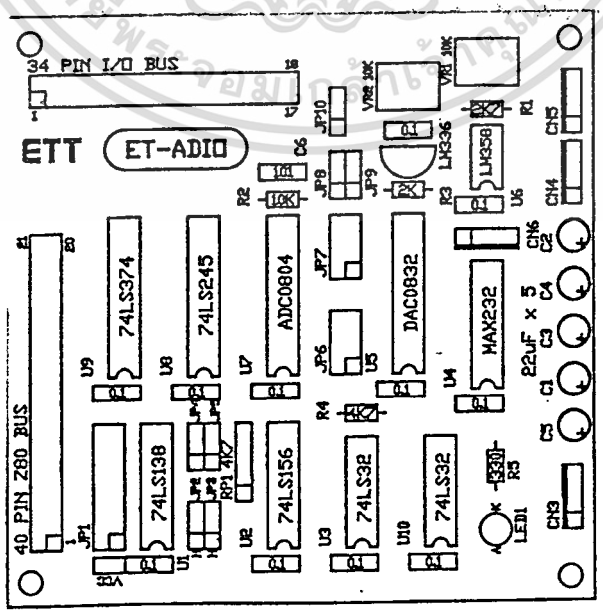
หน้าที่การทำงานของบอร์ดอินพุทและเอ๊าท์พุท

ET-ADIO BOARD เป็นบอร์ดอินพุท เอ๊าต์พุท อีกบอร์ดหนึ่งที่ทางทีมงานอีทีที ได้พัฒนาขึ้นเพื่อสนับสนุนการทำงานของบอร์ดควบคุมต่าง ๆ ของอีทีที ที่มีอยู่ให้สามารถเลือกใช้งานระบบอินพุท เอ๊าต์พุท ทั้งแบบแอนะล๊อก และดิจิตอล ได้ตามความต้องการอย่างเหมาะสม เป็นการเพิ่มประสิทธิภาพและสนับสนุนการใช้งานของบอร์ดควบคุมต่าง ๆ ให้เพิ่มประสิทธิภาพมากขึ้นด้วย

ลักษณะทั่ว ๆ ไปของ ET-ADIO BOARD เป็นบอร์ดอินพุท เอ๊าต์พุทขนาดเล็ก ใช้งานง่าย เหมาะแก่การริเริ่มศึกษาและทำความเข้าใจเกี่ยวกับระบบอินพุท เอ๊าต์พุท ทั้งแบบแอนะล๊อกและดิจิตอลในลักษณะต่าง ๆ เพื่อให้เกิดความรู้ความเข้าใจและเกิดแนวคิดในการที่จะนำไประยุกต์ใช้งานด้านต่าง ๆ อย่างเหมาะสม และมีประสิทธิภาพสูงสุด

ส่วนประกอบที่สำคัญ ๆ ของ ET-ADIO BOARD

- มีภาคอินพุทแบบดิจิตอล จำนวน 8 บิต
- มีภาคเอ๊าต์พุทแบบดิจิตอล จำนวน 8 บิต
- มีภาคอินพุทแบบแอนะล๊อกขนาด 8 บิต หรือ 10 บิต จำนวน 1 แชลแนล
- มีภาคเอ๊าต์พุทแบบแอนะล๊อกขนาด 8 บิต หรือ 12 บิตจำนวน 1 แชลนอล
- มีวงจรถอดรหัสตำแหน่งของพอร์ตใช้งานได้อย่างอิสระถึง 256 พอร์ต



รูปแสดงโครงสร้างของ ET-ADIO BOARD

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานของวงจรภาคอินพุทแบบดิจิตอล

วงจรภาคอินพุทแบบดิจิตอลนี้ ใช้ไอซี 74LS245 เป็นตัวรับสัญญานอินพุททั้ง 8 บิต โดยสัญญานอินพุทแบบดิจิตอลนี้ ควรเป็นสัญญานในระดับ TTL โดยมีจุดรับสัญญานอินพุทผ่านเข้ามาทาง CONNECTOR CN2 (IP0-IP7) แต่บน ET-ADIO BOARD นี้มีวงจรรับสัญญานอินพุทที่มีขนาดสูงกว่าระดับสัญญาน TTL โดยสามารถรับสัญญานอินพุทได้สูงสุดถึง +20V จำนวน 2 บิต โดยใช้ไอซี MAX232 ทำการแปลงสัญญานอินพุทให้มีขนาดลดลงอยู่ในระดับของสัญญาน TTL โดยมีหลักการทำงานดังนี้คือ

ถ้าสัญญานอินพุทที่รับเข้ามามีค่าเป็นลบ (สูงสุด -20V) ไอซี MAX232 จะแปลงสัญญานให้เป็นโลจิกสูง ("1")

ถ้าสัญญานอินพุทที่รับเข้ามามีค่าเป็นบวก (สูงสุด -20V) ไอซี MAX232 จะแปลงสัญญานให้เป็นโลจิกต่ำ ("0")

โดยสัญญานอินพุททั้งสองนี้ต่อผ่านเข้ามาทางขั้วต่อ CONNECTOR CN6 และสัญญานอินพุทที่แปลงเป็นสัญญานในระดับ Ttl แล้วจะต่อไปยังขั้วต่อ CONNECTOR CN2 (IP0 และIP1) ของไอซี 74LS245 อีกทีหนึ่ง ผู้ใช้สามารถเลือกใช้ได้ตามความต้องการ

การทำงานของวงจรภาคเอ๊าต์พุทแบบดิจิตอล

วงจรภาคเอ๊าต์พุทแบบดิจิตอลนี้ใช้ไอซี 74LS374 เป็นตัวส่งสัญญานดิจิตอลทั้ง 8 บิต ผ่านออกเอ๊าต์พุทที่ตำแหน่ง CONNECTOR CN2 (OP0-OP7) ซึ่งสัญญานเอ๊าต์พุทที่ได้จะมีการแลตช์สัญญานไว้ด้วย และเช่นเดียวกันวงจรภาคนี้ยังมีวงจรแปลงขนาดของสัญญานเอ๊าต์พุทในระดับของสัญญาน TTL ให้มีขนาดสูงขึ้นโดยใช้ไอซี MAX232 ทำหน้าที่ดังกล่าว โดยมีหลักการทำงานตรงกันข้ามกับวงจรภาคอินพุทแบบดิจิตอล คือ

ถ้าสัญญานจากไอซี 74LS374 เป็นสัญญานโลจิกต่ำ ("0") ไอซี MAX232 จะแปลงสัญญานให้มีขนาดประมาณ +8V.

ถ้าสัญญานจากไอซี 74LS374 เป็นสัญญานโลจิกสูง("1") ไอซี MAX232 จะแปลงสัญญานให้มีขนาดประมาณ -8V. สัญญานเอ๊าต์พุทที่แปลงได้ทั้ง 2 บิตนี้ เหมาะที่จะใช้ส่งสัญญานไปในสายที่มีระยะทางไกล ๆ ได้ดี โดยสัญญานจะต่อออกไปยังขั้วต่อ CONNECTOR CN6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานของวงจรภาคอินพุทแบบแอนะล็อก

ภาคอินพุทแบบแอนะล็อกนี้ใช้ไอซี ADC (ANALOG TO DIGITAL CONVERTER) ทำหน้าที่เปลี่ยนสัญญานแอนะล็อกให้เป็นสัญญานทางดิจิตอล ซึ่งสามารถเลือกใช้ได้ 2 ขนาดคือ ADC0804 (ADC ขนาด 8บิต) หรือ (ADC 1001 (ADC ขนาด 10บิต) ซึ่งไอซี ADC นี้สามารถรับสัญญานอินพุทแบบแอนะล็อกได้โดยตรงสูงถึง +5V. หรือหากต้องการให้รับอินพุทได้สูงกว่า +5V. แล้วก็สามารถทำได้ โดยการเพิ่มวงจรขยายย่านวัดเข้าไปทางภาคหน้าของวงจรนี้ได้

หลักการทำงานของวงจร

ไอซี ADC จะทำหน้าที่เปลี่ยนสัญญานแอนะล็อกให้เป็นสัญญานทางดิจิตอล โดยใช้หลักการสุ่มรับสัญญานแอนะล็อกทางอินพุท (SAMPLING) แล้วนำขนาดของสัญญานแอนะล็อกนั้น มาเปรียบเทียบกับค่าแรงดันอ้างอิง ของวงจรแล้วจึงเปลี่ยนเป็นค่าข้อมูลแบบดิจิตอล ซึ่งความเร็วในการสุ่มขนาดสัญญาน (SAMPLING RATE) สามารถกำหนดได้จากความถี่ของสัญญานนาฬิกา ของวงจร ซึ่งค่าความถี่ของสัญญานนาฬิกานี้สามารถกำหนดได้โดยวงจร R2 และ C5 โดยมีความสัมพันธ์กันดังนี้คือ

$$F = 1 / (1.1 RC)$$

เมื่อ F คือ ค่าความถี่ของการสุ่มขนาดของสัญญานของไอซี ADC

R คือ ค่าความต้านทานของ R2 ในวงจร

C คือ ค่าความจุของ C5 ในวงจร

ความสัมพันธ์ของอินพุทและเอ้าต์พุทของไอซี ADC

ดังได้กล่าวแล้วว่า ไอซี ADC จะใช้การเปรียบเทียบขนาดของสัญญานอินพุท กับแรงดันอ้างอิงของวงจรแล้วเปลี่ยยเป็นข้อมูลแบบดิจิตอล ซึ่งมีความสัมพันธ์ดังนี้คือ

- ถ้าค่าของขนาดสัญญานแอนะล็อกอินพุทเท่ากับขนาดของแรงดันอ้างอิง จะได้ข้อมูลมีค่าเป็นครึ่งหนึ่ง ของข้อมูลสูงสุด
- ถ้าค่าของขนาดสัญญานแอนะล็อกอินพุทมีค่าเป็น 2 เท่าของแรงดันอ้างอิง จะได้ข้อมูลมีค่าสูงสุด

ซึ่งขนาดของสัญญานอินพุทของวงจรถูกกำหนดโดยค่าตัวแปรต่าง ๆ ดังนี้คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. แรงดันอ้างอิงของวงจร (Voltage Reference หรือ Vref/2)

แรงดันอ้างอิงของวงจรนี้เป็นจุดอ้างอิงมาตราฐานของวงจรเป็นตัวกำหนดความเที่ยงตรงในการเปลี่ยนขนาดของสัญญานแอนะล็อกให้เป็นสัญญานดิจิตอล ซึ่งวงจรต้องการแรงดันอ้างอิงที่มีความเที่ยงตรงสูง เพราะหากแรงดันอ้างอิงของวงจรไม่มีความเที่ยงตรงแน่นอน มีการเปลี่ยนแปลงแล้ว การทำงานของวงจรก็จะเปลี่ยนแปลงตามด้วย นั่นก็คือจะส่งผลให้ค่าของสัญญานข้อมูลดิจิตอลที่ได้ ก็จะเปลี่ยนแปลงขึ้น ๆ ลง ๆ ตามไปด้วย ถึงแม้ว่าสัญญานอินพุทจะมีขนาดคงที่ก็ตาม และแรงดันอ้างอิงของวงจรนี้ยังเป็นตัวกำหนดค่าสูงสุดของสัญญานอินพุทบวกของวงจร โดยค่าสัญญานอินพุทบวกของวงจรจะถูกกำหนดให้มีค่าสูงสุด ประมาณ 2 เท่าของแรงดันอ้างอิงมีจุดเริ่มต้นที่สัญญานอินพุทลบของวงจรเสมอ ซึ่งบน ET-ADIO BOARD นั้นใช้ไอซี LM336 เป็นตัวสร้างสัญญานแรงดันอ้างอิง ขนาด 2.50V ซึ่งเป็นค่าแรงดันอ้างอิงที่ทำให้ไอซี ADC รับสัญญานอินพุทได้โดยตรงสูงสุด +5V แต่ผู้ใช้สามารถเลือกใช้แรงดันอ้างอิงให้กับวงจรได้ 2 แบบคือ

1.1 แรงดันอ้างอิงจากตัวบอร์ดเอง มีขนาด 2.50V

1.2 แรงดันอ้างอิงจากภายนอก ซึ่งกำหนดโดยผู้ใช้เอง โดยต่อเข้ามาทางขั้วต่อ CONNECTOR CN4 ซึ่งแรงดันอ้างอิงนี้ ต้องมีขนาดระหว่าง 0.0V-2.5V

2. สัญญานอินพุทลบ (Voltage Input- หรือ VI-)

แรงดันอินพุทลบนี้ เป็นตัวกำหนดค่าอินพุทด้านต่ำของวงจร หรือจุดเริ่มต้นของการตรวจรับสัญญานอินพุท ของวงจรนั่นเอง โดยสัญญานอินพุทลบนี้สามารถกำหนดได้ให้มีค่าไม่เกินแรงดันอ้างอิงสูงสุดของวงจร (2.50V) ซึ่งบน ET-ADIO BOARD นั้นผู้ใช้สามารถกำหนดสัญญานอินพุทลบของวงจรได้ 2 แบบ ตามความต้องการคือ

2.1 สัญญานอินพุทลบจากตัวบอร์ดเอง โดยกำหนดไว้ในระดับกราวด์ของวงจรหรือ 0.0V นั่นเอง ดังนั้นหากผู้ใช้เลือกใช้สัญญานอินพุทลบจากตัวบอร์ดเอง ก็จะเป็นการกำหนดให้วงจรเริ่มตรวจรับสัญญานจากระดับแรงดัน 0.0V จนถึงระดับแรงดันประมาณ 2 เท่า ของแรงดันอ้างอิงของวงจนนั่นเอง

2.2 สัญญานอินพุทลบจากภายนอก ซึ่งจุดรับสัญญานจะต่อผ่านเข้ามาทางขั้วต่อ CONNECTOR CN 4 และสัญญานอินพุทลบนี้จะถูกกำหนดโดยผู้ใช้เอง โดยต้องกำหนดให้มีค่าไม่เกินแรงดันอ้างอิงของวงจรเสมอ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. สัญญานอินพุทบวก (Voltage Input+ หรือ VI+)

แรงดันอินพุทบวกนี้ ก็คือค่าแรงดันอินพุทของวงจรนั่นเอง ซึ่งต้องกำหนดให้มีขนาดสูงสุดไม่เกินกว่าค่าที่วงจร ADC รับได้ ซึ่งต้องเป็นค่าแรงดันที่มีค่าอยู่ในช่วงบวกด้วย เนื่องจากไอซี ADC ไม่สามารถรับสัญญานอินพุทที่มีค่าเป็นลบได้ สัญญานอินพุทนี้จะมีค่าสูงสุดประมาณ 2 เท่าของแรงดันอ้างอิงโดยมีจุดเริ่มต้นที่สัญญานอินพุทลบของวงจร และค่าสูงสุดของสัญญานอินพุทนี้ต้องมีค่าต่ำกว่า +0.5V ด้วยเสมอ

ตัวอย่าง

เราใช้ไอซี ADC ขนาด 8 บิต (ADC0804)

ให้แรงดันอ้างอิงของวงจร (Vref/2) มีค่าเป็น 2.50V

ให้แรงดันอินพุทลบของวงจร (VI-) มีค่าเป็น 0.0V จะได้ว่า

- แรงดันอินพุทลบของวงจร (VI+) ต้องมีค่าอยู่ระหว่าง 0.0V - 5.0V
- ถ้าขนาดสัญญานแอนะล็อกอินพุทมีค่าเป็น 0.0V จะได้ค่าของข้อมูลเอ๊าต์พุทเป็น 00H
- ถ้าขนาดสัญญานแอนะล็อกอินพุทมีค่าเป็น 2.5V จะได้ค่าของข้อมูลเอ๊าต์พุทเป็น 7FH
- ถ้าขนาดสัญญานแอนะล็อกอินพุทมีค่าเป็น 5.0V จะได้ค่าของข้อมูลเอ๊าต์พุทเป็น FFH

ตัวอย่าง

เราใช้ไอซี ADC ขนาด 8 บิต (ADC0804)

ให้แรงดันอ้างอิงของวงจร (Vref/2) มีค่าเป็น 1.50V

ให้สัญญานอินพุทลบของวงจร (VI-) มีค่าเป็น 0.50V จะได้ว่า

- แรงดันอินพุทบวกของวงจร (VI+) ต้องมีค่าอยู่ระหว่าง 0.50V - 3.50V
- ถ้าขนาดสัญญานแอนะล็อกอินพุทเป็น 0.5V จะได้ค่าของข้อมูลเอ๊าต์พุทเป็น 00H
- ถ้าขนาดสัญญานแอนะล็อกอินพุทมีค่าเป็น 2.0V จะได้ค่าของข้อมูลเอ๊าต์พุทเป็น 7FH
- ถ้าขนาดสัญญานแอนะล็อกอินพุทมีค่าเป็น 3.5V จะได้ค่าของข้อมูลเอ๊าต์พุทเป็น FFH

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานของวงจรภาคเอ๊าต์พุทแบบแอนะล็อก

ภาคเอ๊าต์พุทแบบแอนะล็อกนี้ ใช้ไอซี DAC (DIGITAL TO CONVERTER) เป็นส่วนประกอบที่สำคัญของวงจร โดยไอซี DAC จะทำหน้าที่ตรงข้ามกับไอซี ADC กล่าวคือ มันจะทำหน้าที่เปลี่ยนสัญญานข้อมูลแบบดิจิตอล ให้เป็นสัญญานเอ๊าต์พุทแบบแอนะล็อก ซึ่งใน ET-ADIO BOARD ผู้ใช้สามารถเลือใช้ไอซี DAC ได้ 2 ขนาด คือ DAC0832 ซึ่งเป็นไอซี DAC ขนาด 8 บิต หรือ DAC 232 ซึ่งเป็นไอซี DAC ขนาด 2 บิต ซึ่งไอซี DAC นี้จะให้สัญญานเอ๊าต์พุทออกมาเป็นแบบแอนะล็อก แต่ขนาดของสัญญานที่ได้ยังมีขนาดต่ำไม่สามารถนำไปใช้ควบคุมการทำงานของวงจรอื่นได้โดยตรง ต้องมีวงจรขยายสัญญานเอ๊าต์พุทให้มีขนาดสูงขึ้นก่อนนำไปใช้

หลักการทำงานของวงจร

ไอซี DAC จะทำหน้าที่นำข้อมูลอินพุทแบบดิจิตอลมาเปรียบเทียบกับแรงดันอ้างอิงของวงจร แล้วเปลี่ยนเป็นสัญญานแอนะล็อกออกทางเอ๊าต์พุท ซึ่งขนาดของสัญญานเอ๊าต์พุทที่ได้จะมีความสัมพันธ์กับขนาดของข้อมูลอินพุทแบบเป็นสัดส่วนเชิงเส้น (LINEAR) โดยขนาดของสัญญานเอ๊าต์พุทนั้น สามารถแบ่งขนาดออกได้เป็นช่วง ๆ (STEP) ตามขนาดของไอซี DAC ที่ใช้ค่า STEP นี้จะเป็นตัวแสดงถึงความละเอียดของสัญญานเอ๊าต์พุทของวงจร DAC ในแต่ละช่วงของการเปลี่ยนแปลงของจำนวนข้อมูลอินพุท ว่าเมื่อค่าข้อมูลเพิ่มขึ้น 1 ระดับขนาดของสัญญานเอ๊าต์พุทเพิ่มขึ้นเท่าไร ถ้าไอซี DAC มีจำนวนบิทมาก ค่าความละเอียดก็จะมากถ้าไอซี DAC มีจำนวนบิทน้อย ค่าความละเอียดก็จะมีค่าน้อยตามไปด้วย

ความสัมพันธ์ของอินพุทและเอ๊าต์พุทของไอซี DAC

ดังได้กล่าวแล้วว่าไอซี DAC จะใช้การเปรียบเทียบจำนวนของข้อมูลอินพุทแบบดิจิตอลกับแรงดันอ้างอิงของวงจรแล้วเปลี่ยนขนาดของข้อมูลอินพุทนั้นกลับเป็นสัญญานแบบแอนะล็อกส่งออกทางเอ๊าต์พุท ซึ่งขนาดของข้อมูลอินพุท กับขนาดของสัญญานเอ๊าต์พุทมีความสัมพันธ์ดังนี้คือ

- ถ้าค่าของข้อมูลอินพุทมีค่าสูงสุด (เป็นโลจิก "1" ทั้งหมดทุกบิท) จะได้ขนาดของสัญญานแอนะล็อกเอ๊าต์พุทมีค่าเป็น2 เท่าของแรงดันอ้างอิงของวงจร
- ถ้าค่าของข้อมูลอินพุทมีค่าเป็นครึ่งหนึ่งของข้อมูลสูงสุด จะได้ขนาดสัญญานแอนะล็อกเอ๊าต์พุท มีค่าเท่ากับค่าของแรงดันอ้างอิงของวงจร

ซึ่งขนาดของสัญญานเอ๊าต์พุทของวงจรถูกกำหนดโดยค่าตัวแปรต่าง ๆ ดังนี้คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. แหล่งจ่ายไฟเลี้ยงวงจร DAC

ค่าของแรงดันของแหล่งจ่ายไฟที่ป้อนให้วงจร DAC นี้จะเป็นตัวกำหนดขนาดสูงสุดของสัญญานแอนะล๊อกเอ๊าต์พุทที่สามารถทำได้โดยผู้ใช้สามารถเลือกใช้ได้ 2 ทางคือ

1.1 แหล่งจ่ายไฟเลี้ยงจากตัวบอร์ดเอง เป็นแรงดันไฟตรงแบบ บวก ลบ และ กราวด์ เพื่อป้อนให้วงจร DAC ซึ่งมีค่าประมาณ 10 VDC ซึ่งแรงดันในส่วนนี้สร้างขึ้นจากไอซี MAX232 ในวงจรซึ่งจะจ่ายแรงดันนี้ให้กับวงจร DAC โดยตรงอยู่แล้ว แต่การเลือกใช้แหล่งจ่ายไฟในตัวบอร์ดนี้ มีข้อจำกัดอย่างหนึ่งคือ จะได้ขนาดของสัญญานเอ๊าต์พุทสูงสุดมีค่าได้ +15VDC

2. แรงดันอ้างอิงของวงจร (Voltage Reference หรือ Vref/2)

แรงดันอ้างอิงของวงจรนี้ เป็นจุดอ้างอิงมาตราฐานของวงจรเป็นตัวกำหนดความเที่ยงตรงในการเปลี่ยนขนาดของสัญญานข้อมูลแบบดิจิตอลให้เป็นสัญญานแอนะล๊อก ซึ่งวงจรต้องการแรงดันอ้างอิงที่มีความเที่ยงตรงสูง เพราะหากแรงดันอ้างอิงของวงจรไม่มีความเที่ยงตรงแน่นอน มีการเปลี่ยนแปลงแล้ว การทำงานของวงจรก็จะเปลี่ยนแปลงตามด้วย นั่นก็คือจะส่งผลให้ค่าของสัญญานแอนะล๊อกเอ๊าต์พุทที่ได้เปลี่ยนแปลงขึ้น ๆ ลง ๆ ตามไปด้วย ถึงแม้ว่าค่าของข้อมูลอินพุทที่ป้อนให้ไอซี DAC จะมีขนาดคงที่ก็ตาม และแรงดันอ้างอิงของวงจรนี้ยังเป็นตัวกำหนดสัดส่วนของข้อมูลอินพุทและสัญญานเอ๊าต์พุทอีกด้วย ซึ่งบน ET-ADIO BOARD นั้นใช้ไอซี LM336 เป็นตัวสร้างสัญญานแรงดันอ้างอิง ขนาด 2.50V ซึ่งเป็นค่าแรงดันอ้างอิงชุดเดียวกันกับวงจรที่สร้างแรงดันอ้างอิงให้ไอซี ADC ซึ่งแรงดันอ้างอิงนี้สามารถเลือกใช้ได้ 2 ทางคือ

2.1 แรงดันอ้างอิงภายในตัวบอร์ดเองใช้ไอซี LM336 เป็นวงจรควบคุมขนาดของแรงดันอ้างอิงให้วงจร ซึ่งเป็นวงจรชุดเดียวกันกับที่ป้อนให้วงจร ADC มีขนาด 2.50V

2.2 แรงดันอ้างอิงจากภายนอก ซึ่งต่อผ่านเข้ามาทางขั้วต่อ CONNECTOR CN5 โดยแรงดันอ้างอิงนี้ ต้องเป็นไฟตรงมีค่าสูงสุดได้ไม่เกิน +10V และต้องควบคุมให้มีค่าน้อยกว่าแรงดันของแหล่งจ่ายไฟที่ป้อนให้กับวงจร DAC เสมอ

3. ค่าเกณฑ์การขยายของวงจร OP-AMP

เนื่องจากสัญญานแอนะล๊อกเอ๊าต์พุทที่ได้จากไอซี DAC นั้นมีขนาดต่ำ ไม่สามารถนำไปใช้งานได้โดยตรง จึงต้องใช้วงจรขยายขนาดของสัญญานให้สูงขึ้นดังนั้นเกณฑ์การขยายของวงจรจึงเป็นตัวกำหนดอัตราส่วนของขนาดสัญญานสูงสุดที่ต้องการด้วย

เอกสารนี้ ... ให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. ค่าของข้อมูลอินพุท (DATA)

ค่าของข้อมูลอินพุทนี้เป็นตัวกำหนดขนาดของสัญญานเอ๊าต์พุทโดยตรง ซึ่งค่าของตัวแปรอื่น ๆ นั้น เรากำหนดเพียงครั้งเดียวแล้วกำหนดคงที่ไว้ที่ตำแหน่งนั้นเลย แต่ค่าของข้อมูลอินพุทนี้ สามารถเปลี่ยนแปลงขึ้น-ลง ได้ตลอดเวลาเพื่อควบคุมขนาดของสัญญานเอ๊าต์พุทในช่วงเวลาต่าง ๆ ตามความต้องการ ซึ่งถ้าค่าของข้อมูลมีค่ามาก ก็จะได้ขนาดของสัญญานเอ๊าต์พุทมาก แต่ถ้าค่าของข้อมูลมีค่าน้อยก็จะทำให้ได้ขนาดของสัญญานเอ๊าต์พุทน้อยตามไปด้วย

<u>การกำหนดค่าของข้อมูลเพื่อกำหนดขนาดของสัญญานเอ๊าต์พุท</u>

การที่เราจะสามารถกำหนดค่าของข้อมูลอินพุทของไอซี DAC เพื่อให้ได้ขนาดของสัญญานเอ๊าต์พุท ตามความต้องการของเรานั้น ก่อนอื่นเราต้องทราบความสามารถของไอซี DAC เสียก่อนว่าสามารถให้ค่าสัญญานเอ๊าต์พุทต่อ STEP เป็นเท่าไรเสียก่อน แล้วจึงหาว่าหากเราต้องการได้ค่าสัญญานเอ๊าต์พุทขนาดเท่านี้แล้วต้องใช้กี่ STEP จึงจะได้สัญญานตามต้องการซึ่งค่า STEP นี้ สามารถกำหนดได้จากจำนวนบิทข้อมูลของไอซี DAC ที่เราใช้ในวงจร

ถ้าเราใช้ไอซี DAC ขนาด 8 บิท จะได้ค่าของข้อมูลอยู่ระหว่าง 00H-FFH หรือ 0-255 นั่นก็คือ ไอซี DAC สามารถให้สัญญานเอ๊าต์พุทเป็นช่วง ๆ ทั้งหมด 256 ช่วง (0-255) นั่นเอง

ถ้าเราใช้ไอซี DAC ขนาด 12 บิท จะได้ค่าของข้อมูลอยู่ระหว่าง 000H-FFFH หรือ 0-1023 นั่นก็คือ ไอซี DAC สามารถให้สัญญานเอ๊าต์พุทเป็นช่วง ๆ ทั้งหมด 1024 ช่วง (0-1023) นั่นเอง

ซึ่งเมื่อเราทราบว่าไอซี DAC มีค่า STEP ทั้งหมดเท่าใดแล้ว ก็สามารถทราบได้ว่าใน 1 STEP จะได้ขนาดของสัญญานเอ๊าต์พุทต่อ 1 STEP เป็นเท่าใดโดยสามารถหาได้จากสูตร

STEP = ANALOG MAX / DAC STEP

เมื่อ STEP คือ ค่าของขนาดสัญญานแอนะล๊อกเอ๊าต์พุทต่อ 1 ช่วงข้อมูล

ANALOG MAXIMUM คือ ค่าของขนาดสัญญานแอนะล๊อกเอ๊าต์พุทสูงสุดที่ใช้ในวงจร

DAC คือ ค่าของจำนวน STEP ของไอซี DAC ที่ใช้ในวงจรโดย ถ้าใช้ไอซี DAC0832 (DAC ขนาด 8 บิต) DAC STEP มีค่าเป็น 256 ถ้าใช้ไอซี DAC1232 (DAC ขนาด 12 บิต) DAC STEP มี ค่าเป็น 1024

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และเราจะสามารถหาค่าข้อมูลได้จากสูตร

DATE = Vout / STEP

เมื่อ DATE คือ ค่าของข้อมูลที่ต้องการหา

Vout คือ ค่าของขนาดสัญญานแอนะล๊อกเอ๊าต์พุทที่ต้องการ

STEP คือ ค่าของขนาดสัญญานแอนะล๊อกเอ๊าต์พุทต่อ1 ช่วงข้อมูล

ตัวอย่าง

ใช้ไอซี DAC ขนาด 8 บิต (DAC0832)

ใช้แรงดันอ้างอิงของวงจร (Vref/2) เป็น 2.50V

ให้สัญญานเอ๊าต์พุทมีขนาดสูงสุด +5V

จะได้ค่าของสัญญานเอ๊าต์พุทต่อ 1 STEP เป็น

STEP = 5/256

= 0.02V

จะเห็นได้ว่าเราได้ค่าของสัญญานเอ๊าต์พุทต่อ STEP มีค่าประมาณ 0.02V หรืออาจกล่าวได้ว่า หากเราให้ข้อมูลอินพุทกับวงจรด้วยค่า 01H(00000001B)เราจะได้สัญญานเอ๊าต์พุทมีขนาดประมาณ 0.02V ดังนั้นหากเราต้องการได้ขนาดของสัญญานเอ๊าต์พุทมีค่าเป็น 2.50V เราต้องให้ค่าข้อมูลกับวงจรเป็น

DATE = 2.50/0.02

= 128 (7FH หรือ 01111111B)

แต่เนื่องจาก ET-ADIO BOARD เป็นบอร์ดที่มีการเชื่อมต่อแบบ 8 บิต คือมีจำนวนของสัญญานข้อมูลทั้งหมด 8 เส้น หากผู้ใช้ต้องการใช้งานไอซี DAC เป็นแบบ 12 บิต (DAC1232) การส่งข้อมูลให้ไอซี DAC ไม่สามารถส่งครั้งละ 12 บิตได้ ต้องทำการแยกส่งข้อมูลให้กับไอซี DAC 2 ครั้ง โดยมีวิธีการดังนี้คือ

1. ส่งข้อมูล 8 บิตล่าง (D0-D7) ให้กับพอร์ตควบคุมไอซี DAC ที่เป็นพอร์ตเบอร์คู่ (พอร์ตที่ A0 มีค่าเป็น "0")

2. ส่งข้อมูล 4 บิตบน (D8-D11) ให้กับพอร์ตควบคุมไอซี DAC ที่เป็นพอร์ตเบอร์คี่ (พอร์ตที่ A0 มีค่าเป็น "1")

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานของวงจรถอดรหัสตำแหน่งพอร์ต

เนื่องจากบอร์ดควบคุมแต่ละบอร์ดมักมีวงจรที่แตกต่างกัน ดังนั้นตำแหน่งการติดต่อกับส่วนอินพุทเอ้าต์พุท ก็ย่อมต้องมีความแตกต่างกันด้วย ดังนั้น ET-ADIO BOARD จึงต้องมีวงจรถอดรหัสตำแหน่งพอร์ตใช้งานที่สามารถทำการเปลี่ยนแปลง ให้ตรงกับตำแหน่งการเชื่อมต่อกับบอร์ดควบคุมต่าง ๆ เหล่านั้นได้ทุกบอร์ด ซึ่ง ET-ADIO BOARD สามารถที่จะกำหนดเบอร์พอร์ตใช้งานได้ถึง 256 ตำแหน่ง จึงสามารถเชื่อมต่อกับบอร์ดควบคุมต่าง ๆ ได้อย่างไม่มีปัญหา

หลักการทำงานของวงจร

วงจรไอซี 74LS138(U1) 74LS156(U2) 74LS32(U3 และ U10) และ JUMPER (JP1-JP5) ต่อรวมกันเป็นวงจรถอดรหัสตำแหน่งพอร์ต โดยให้ไอซี 74LS138 เป็นตัวกำหนดตำแหน่งพอร์ตชั้นนอกและให้ไอซี 74LS156 เป็นตัวกำหนดตำแหน่งพอร์ตชั้นใน ซึ่งทั้งตำแหน่งของพอร์ตทั้ง 2 ชั้นยังสามารถแบ่งออกเป็น BANK คือ BANK สูง และ BANK ต่ำ โดยมี JUMPER JP1,JP2 ทำหน้าที่เลือกตำแหน่ง BANK ของพอร์ตชั้นนอกว่าจะให้ทำงานใน BANK สูงหรือ BANK ต่ำ ซึ่งตำแหน่งของพอร์ตที่ได้จากวงจรชุดนี้จะถูกต่อไปยัง JUMPER JP5 โดยมีตำแหน่งพอร์ตดังนี้คือ

JUMPER JP5 SET	เบอร์พอร์ตใน BANK ต่ำ	เบอร์พอร์ตใน BANK สูง
SHORT 1-2 ONLY	00 H - 0F H	80 H - 8F H
SHORT 3-4 ONLY	10 H - 1F H	90 H - 9F H
SHORT 5-6 ONLY	20 H - 2F H	A0 H - AF H
SHORT 7-8 ONLY	30 H - 3F H	B0 H - BF H
SHORT 9-10 ONLY	40 H - 4F H	C0 H - CF H
SHORT 11-12 ONLY	50 H - 5F H	D0 H - DF H
SHORT 13-14 ONLY	60 H - 6F H	E0 H - EF H
SHORT 15-16 ONLY	70 H - 7F H	F0 H - FF H

นอกจากนี้ยังมี JUMPER JP3 และ JP4 ทำหน้าที่เลือก BANK ของตำแหน่งเบอร์พอร์ตชั้นในโดยมีตำแหน่งดังนี้คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วงจรที่ทำงานในตำแหน่งพอร์ตนี้	เบอร์พอร์ตใน BANK ต่ำ	เบอร์พอร์ตใน BANK สูง
วงจรภาคเอ๊าต์พุทแบบแอนะล๊อก (DAC)	XX0 H - XX1 H	XX8 H - XX9 H
วงจรภาคอินพุทแบบแอนะล๊อก (ADC)	XX2 H - XX3 H	XXA H - XXB H
วงจรภาคอินพุทแบบดิจิตอล (IN)	XX4 H - XX5 H	XXC H - XXD H
วงจรภาคเอ๊าต์พุทแบบดิจิตอล (OUT)	XX6 H - XX7 H	XXE H - XXF H

ซึ่งค่า XX ที่กำกับอยู่หน้าเบอร์พอร์ตนั้น หมายถึงค่าของเบอร์พอร์ตที่ถูกกำหนดมาแล้วจากส่วนของวงจรถอดรหัสของพอร์ตชั้นนอก

สมมติว่า

ถ้าเรากำหนดตำแหน่งของพอร์ตชั้นนอกไว้ที่ตำแหน่ง 80H - 8FH

กำหนดให้วงจรถอดรหัสชั้นในทำงานใน BANK ต่ำจะได้ว่า

วงจรภาคเอ๊าต์พุทแบบแอนะล๊อก (DAC) มีตำแหน่งพอร์ตควบคุมเป็น 80H - 81H

วงจรภาคอินพุทแบบแอนะล๊อก (ADC) มีตำแหน่งพอร์ตควบคุมเป็น 82H - 83H

วงจรภาคอินพุทแบบดิจิตอล (IN) มีตำแหน่งพอร์ตควบคุมเป็น 84H - 85H

วงจรภาคเอ๊าต์พุทแบบดิจิตอล (OUT) มีตำแหน่งพอร์ตควบคุมเป็น 86H - 87H

แต่ถ้าเรากำหนดตำแหน่งของพอร์ตชั้นนอกไว้ที่ตำแหน่ง 80H - 8FH

กำหนดให้วงจรถอดรหัสชั้นในทำงานใน BANK สูงจะได้ว่า

วงภาคเอ๊าต์พุทแบบแอนะล๊อก (DAC) มีตำแหน่งพอร์ตควบคุมเป็น 88H - 89H

วงจรภาคอินพุทแบบแอนะล๊อก (ADC) มีตำแหน่งพอร์ตควบคุมเป็น 8AH - 8BH

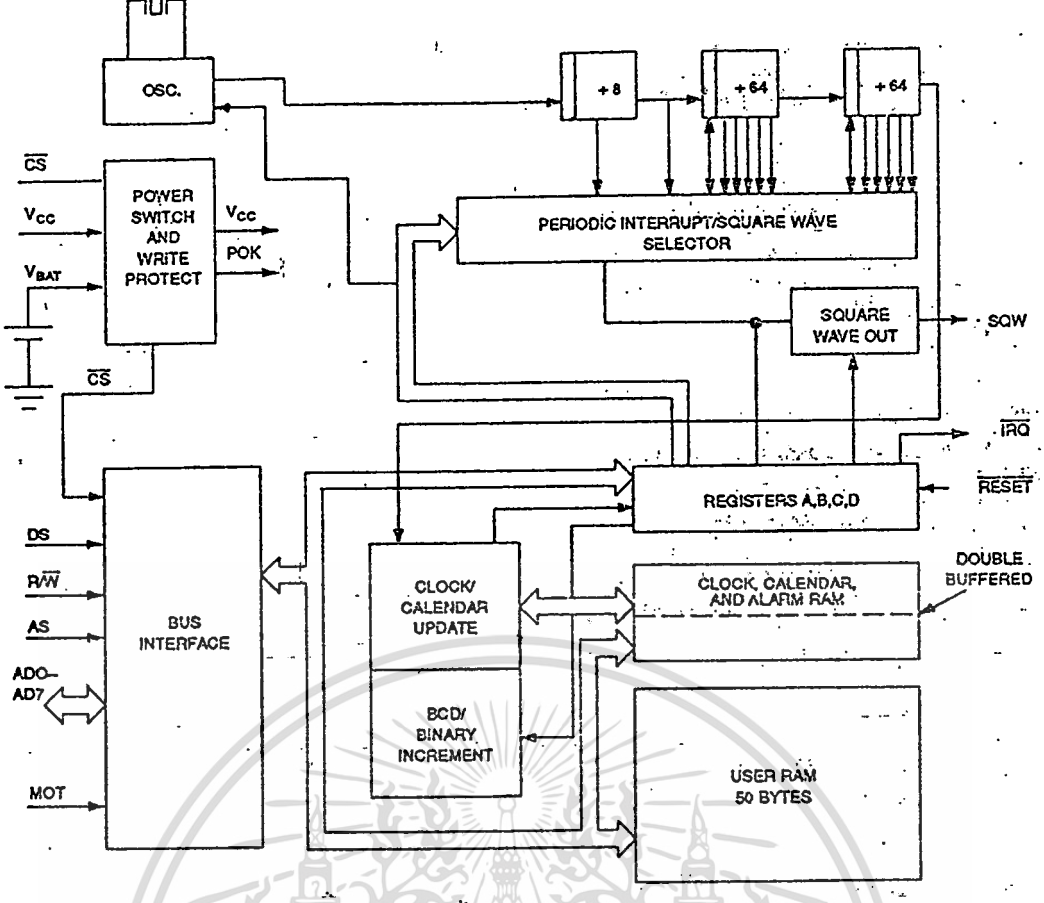
วงจรภาคอินพุทแบบดิจิตอล (IN) มีตำแหน่งพอร์ตควบคุมเป็น 8CH - 8DH

วงจรภาคเอ๊าต์พุทแบบดิจิตอล (OUT) มีตำแหน่งพอร์ตควบคุมเป็น 8EH - 8FH

<u>หน้าที่และการทำงานของสัญญาณเวลาจริง (Real Time clock)</u>

เนื่องจากการจำลองขบวนการเป็นการทำงานในลักษณะของโปรแกรมการคำนวณ จึงต้องใช้เวลาในการประมวลผลมาก และการส่งสัญญาณออกทางจอแสดงผล และทางเอ๊าต์พุทนั้นจะต้องมีเวลาที่แน่นอน (Sample Time) จึงเป็นเหตุผลที่ต้องนำ Real Time clock มาใช้งานในการ Sampling ข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปแสดงบล๊อคไดอะแกรมของ DS1287

DS1287 ออกแบบมาเพื่อใช้เป็น Real Time clock ซึ่งมีวิธีการใช้งานง่ายและสะดวก มีแบตตเตอรี่ในตัวเอง (lithium energy source) มีฟังก์ชั่นให้เลือกเช่น time-of-day, clock และ Alarm หรือใช้เป็นปฏิทิน เป็นชิพ (ship) ที่เหมาะแก่การใช้งานได้หลายประเภท และมีหน่วยความจำภายในตัวเอง 50 bytes สามารถที่จะเป็นแหล่งกำเนิดสัญญาณได้อีกด้วย (Square wave generator)

<u>การใช้งาน DS1287</u>

จากตารางแสดงการเลือกใช้ช่วงเวลาของ DS1287 ซึ่งการ Sampling ของชุดจำลองกระบวนการใช้เวลา 1 วัน .Sce. จึงออกแบบให้ใช้ 500x2 m5 ซึ่งสัญญานนี้จะส่งให้ INT1 ของบอร์ด ET-8032 เพื่อขัดจังหวะให้ชุดจำลองส่งข้อมูลออกทางพอร์ตอนุกรม และทางเอ๊าต์พุทของชุดจำลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SELECT BITS REGISTER A				t_{PI} PERIODIC INTERRUPT RATE	SQW OUTPUT FREQUENCY
RS3	RS2	RS1	RS0		
0	0	0	0	None	None
0	0	0	1	3.90625 ms	256 Hz
0	0	1	0	7.8125 ms	128 Hz
0	0	1	1	122.070 µs	8.192 kHz
0	1	0	0	244.141 µs	4.096 kHz
0	1	0	1	488.281 µs	2.048 kHz
0	1	1	0	976.5625 µs	1.024 kHz
0	1	1	1	1.953125 ms	512 Hz
1	0	0	0	3.90625 ms	256 Hz
1	0	0	1	7.8125 ms	128 Hz
1	0	1	0	15.325 ms	64 Hz
1	0	1	1	31.25 ms	32 Hz
1	1	0	0	62.5 ms	16 Hz
1	1	0	1	125 ms	8 Hz
1	1	1	0	250 ms	4 Hz
1	1	1	1	500 ms	2 Hz

รูป ตารางแสดงการเลือกใช้สัญญานนาฬิกา

RS0-RS1 จะเป็นการ decode ของสัญญานจากบอร์ด ET-8032 เพื่อเลือกใช้สัญญานนาฬิกาของ DS1287

DS (Data strobe or Read Input) เป็นสัญญานการอ่านข้อมูลเพื่อการเซ็ตค่าต่าง ๆ ของ DS1287

CS (Chip select Input) เป็นสัญญานเลือกการติดต่อระหว่าง PCU กับ DS1287

IRQ (Interrupt Request Output) เป็นการใช้สัญญานการขัดจังหวะ โดยการใช้งานผ่านทาง Register C ซึ่งสามารถกำหนดว่าใช้หรือไม่ใช้ได้

RESET (Reset Input) รีเซ็ตระบบของ DS1287

LINEARITY คือค่าความคงที่ในการเปลี่ยนระดับของสัญญานมีหน่วยเป็นเปอร์เซ็นต์ปกติ D/A ควรจะมีค่าการเปลี่ยนระดับเป็นเส้นตรง แต่ในความจริงอาจจะมีการเบี่ยงเบนเกิดขึ้นได้ ในทางจุดปกติไม่ควรมีค่ามากกว่า +12LBS ซึ่งก็คือค่าเปอร์เซ็นต์ของ RESOLUTION หารด้วย 2 คือ 0.39122 = 0.019%

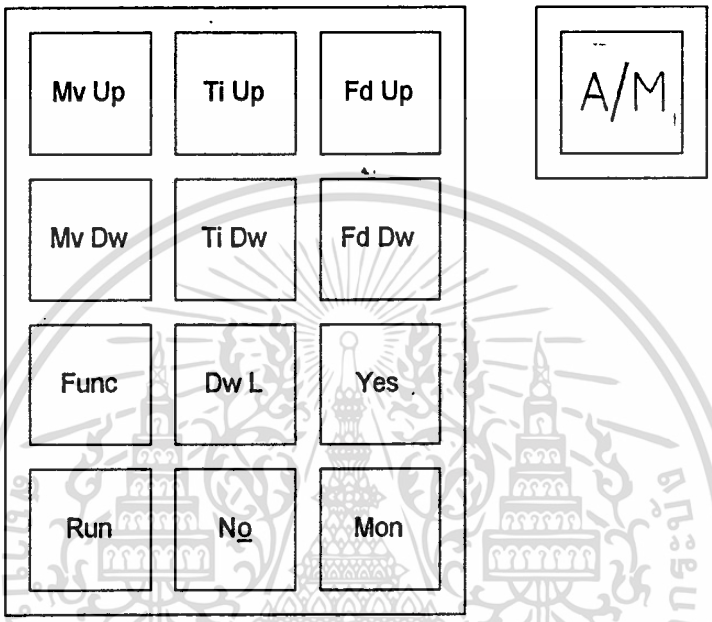
SETTING TIME คือ ค่าเวลาที่ใช้ในการเปลี่ยนระดับสัญญานเท่ากับ 1/2 ตัวอย่างเช่น ไอซีเบอร์ DAC 0808 จะมีค่าเท่ากับ 150 ns ค่ามีความสำคัญต่อความเร็วในการเปลี่ยนหรืออีกนัยหนึ่งก็คือความถี่ของสัญญาน ANALOG นั่นเอง

วงจรพื้นฐานของ D/A จะมีอยู่ 2 แบบคือ WEIGHTERED - RESISTOR D/A และ R2R D/A ในการทดลอง อย่างไรก็ตามปัจจุบันมักจะใช้พวกไอซีสำเร็จรูป ซึ่งง่ายต่อการใช้งานและมีประสิทธิภาพดี

เอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้าที่การใช้งานของคีย์บอร์ด

คีย์บอร์ดจะจำแนกเป็น 2 ส่วน ในส่วนแรกจะเป็นการใช้งานของการใช้เป็นสัญญานรบกวนของระบบ และอีกส่วนหนึ่งจะเป็นการใช้ของการสั่งงาน



รูป ลักษณะของแป้นคีย์บอร์ด

หน้าที่และการทำงานของคีย์บอร์ด

A/M: ใช้ในการเลือกโหมด Auto หรือโหมด Munual เพื่อเลือกการใช้งาน Mv โดยจะมีสัญญาน LED ซึ่งสีแดง แสดงว่าเป็นโหมด Auto และสีเขียวแสดงโหมด Munual

MvUp : ใช้เพิ่มค่า Mv

MvDw : ใช้ลดค่า Mv

TiUp : ใช้เพิ่มค่าอุณหภูมิของเหลว (Temp feed)

TiDw : ใช้ลดค่าอุณหภูมิของเหลว (Temp feed)

FdUp : ใช้เพิ่มค่าการไหล (Flow feed)

FdDw : ใช้ลดค่าการไหล (Flow feed)

Func : เป็นการใช้ฟังก์ชั่นการโหลดข้อมูล

DWL : การโหลดข้อมูล

YES : รับค่าข้อมูล

RUN : เริ่มต้นทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

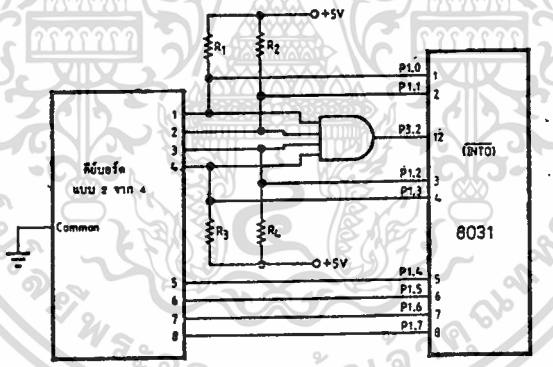
NO : ยกเลิกการทำงาน

MON : สภาวะเริ่มต้น

<u>โปรแกรมสแกนคีย์บอร์ดโดยใช้การอินเตอร์รัพต์</u>

โปรแกรมสแกนคีย์บอร์ดโดยทั่วไปจะเสียเวลาของตัวประมวลผล (CPU) จึงให้มีการใช้อินเตอร์รัพต์ ซึ่งถ้าไม่มีการกดคีย์บอร์ดจะทำให้ไม่มีการอินเตอร์รัพต์เกิดขึ้นเลย ฮาร์ดแวร์ดังแสดงตามรูปจะเป็นการใช้คีย์แบบเซ็ครหัส โดยมี AND เกตต่อเมื่อสร้างสัญญานอินเตอร์รัพต์ที่ INTO

การทำงานของโปรแกรม จะใช้สแกนคีย์ได้ 0-9 คีย์ โดยใช้ขาพอร์ต 3.2 ซึ่งเป็นขา INTO เป็นตัวอินเตอร์รัพต์ ส่วน T0 นั้นจะเป็นเวลาการกระเด้งของคีย์และ T1 เป็นเวลาเมื่อปล่อยคีย์ใน T0 และ T1 นั้นใช้วิธีการอินเตอร์รัพต์เหมือนกัน ส่วนรหัสคีย์ที่กดแต่ละตำแหน่งนั้นจะมีรหัสประจำ ซึ่งจะดูได้จากตาราง



รูป แสดงวงจรสแกนคีย์แบบมีอินเตอร์รัพต์

ตาราง. รหัสประจำคีย์บอร์ดสำหรับโปรแกรม.

หมายเลขคีย์	รหัส
0	EE
1	ED
2	EB
3	E7
4	DE
5	DD
6	DB
7	D7
8	BE
9	BD

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

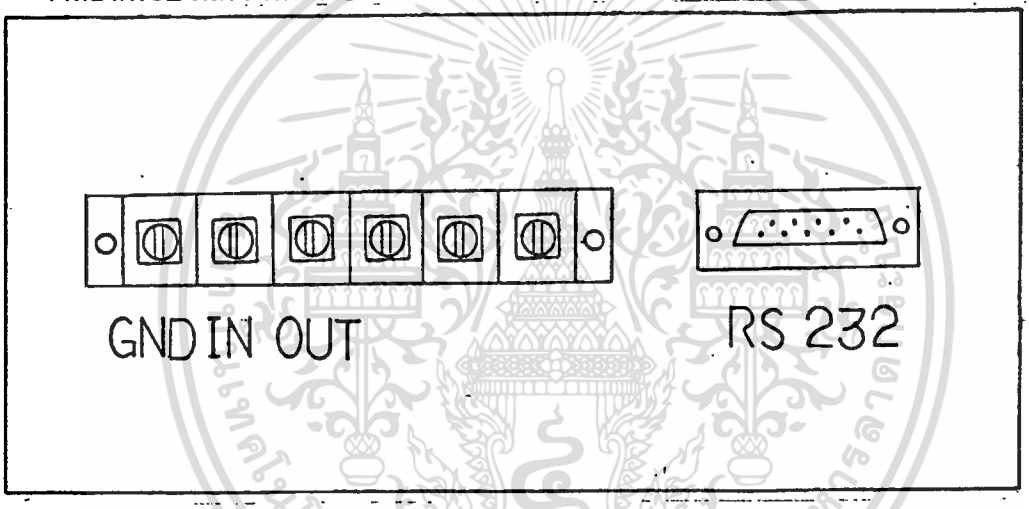
วิธีการใช้งานตัวจำลองกระบวนการ

เริ่มแรกคือการต่อแหล่งจ่ายไฟเข้าให้แก่ตัวจำลองกระบวนการ ต่อสายตามขั้วต่อโดยมีสัญญาณที่สำคัญคือ

1) สัญญาณอินพุท (MV Input) เป็นขั้วสัญญาณอินพุทจากกระบวนการโดยการรับสัญญาณจากตัวควบคุม (PID Controller) มีแรงดันอยู่ระหว่าง 1 - 5 Volts

2) สัญญาณเอาต์พุตจากกระบวนการจำลอง (PV Input) โดยกระบวนการจำลองจะสร้างสัญญาณให้แก่ตัวควบคุม (PID Controller)

3) สัญญาณอนุกรม (Serial Port) เป็นสัญญาณที่ใช้ติดต่อระหว่างตัวจำลองกระบวนการกันเครื่องคอมพิวเตอร์



รูป แสดงหน้าปัดด้านหลังของตัวจำลองกระบวนการ

ขั้นตอนการปฏิบัติในการใช้งานของเครื่องจำลองกระบวนการ

1) เมื่อเริ่มเปิดเครื่อง(จ่ายแรงดันไฟฟ้าเข้าแล้ว) จะต้องมีการโหลดโปรแกรมจากเครื่องคอมพิวเตอร์ สู่ตัวจำลองกระบวนการ

2) กดคีย์สั่งงานของตัวควบคุมกระบวนการ เพื่อให้อยู่ในสภาวะของการ รับข้อมูลจากคอมพิวเตอร์

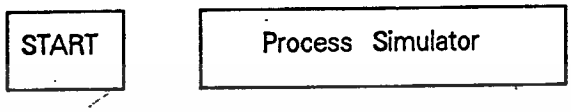
ลักษณะของการสั่งงานมีดังนี้

Func	ตัวจำลองกระบวนการเข้าสู่การเตรียมพร้อมเพื่อการรอ
Func	รับข้อมูลและโปรแกรมจากเครื่องคอมพิวเตอร์โดยผ่าน
Dwl	ทางพอร์ทอนุกรมของเครื่องจำลองกระบวนการ
Yes	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3) โหลดโปรแกรม Simulat จากคอมพิวเตอร์และเข้าสู่เมนูของโปรแกรม เพื่อใช้ในการแสดงผลในรูปแบบของเส้นกราฟ (Line Graph)

4) กดคีย์ START จากตัวอย่างจำลองกระบวนการ ตัวจำลองจะเริ่มต้นในการทำงาน และแสดงผลทาง LCD



Key Board LCD Display

5) การเลือกใช้โหมดการทำงาน สามารถเลือกเป็นแบบอัตโนมัติ (Auto) หรือ ควบคุมด้วยมือ (Manual) โดยการสั่งงานจากคีย์ A/M ซึ่งจะมี LED แสดงค่าของโหมดการทำงานให้ทราบ

○ RED LED = AUTO MODE

○ GREEN LED = MANUAL MODE

MANUAL MODE

เราสามารถที่จะใช้คีย์ MvUp และ MvDw ได้โดยคีย์ที่ใช้จะเป็นการขอขัดจังหวะการทำงานของ CPU

KEY-BOARD LCD Display

หน้าที่การทำงานของคีย์นี้นั้น จะเป็นเพิ่มขึ้นของสัญญาณ MV ซึ่งเป็นสัญญาณเข้าทางอินพุทของเครื่องจำลองกระบวนการ และเป็นการสั่งงานให้มีการเปิดวาล์ควบคุมให้เพิ่มขึ้นด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

KEY BOARD LCD Display

MvDw	MV_Valve => 0.0 %

หน้าที่การทำงานของคีย์ จะเป็นการลดลงของสัญญาณ MV ซึ่งเป็นสัญญาณเข้าทางอินพุทของเครื่องจำลองกระบวนการ และเป็นการสั่งงานให้มีการปิดวาล์วควบคุมให้ลดลงด้วย

AUTO MODE

เราจะไม่สามารถใช้คีย์ MvUp และ MvDw ได้ซึ่งในโหมดนี้เครื่องจำลองกระบวนการจะใช้สัญญาณอินพุทจากวงจรการแปลงสัญญาณอนาลอกไปเป็นแบบดิจิตอล

6) คีย์ที่ทำหน้าที่เป็นตัวรบกวนของระบบ ซึ่งจะแบ่งเป็น 2 ชนิด ด้วยกัน ดังรายละเอียดต่อไปนี้

TiUp	FdUp
TiDw	FdDw

6.1) คีย์ที่ทำหน้าที่เป็นสัญญาณรบกวน อุณหภูมิของของไหล (TEMP feed) จากช่วงของอุณหภูมิ 100 - 200 F

KEY BOARD LCD Display

TiUp	TEMP => 99.9 %

ทำหน้าที่ในการเพิ่มอุณหภูมิของ ของไหล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

KEY BOARD LCD Display

TiDw	TEMP => 0.0 %

ทำหน้าที่ในการลดอุณหภูมิของ ของไหล

6.2) คีย์ที่ทำหน้าที่เป็นสัญญาณรบกวน ของอัตราการไหลของ ของไหล (Flow feed) จากช่วงของอัตราการไหล 15 - 25 ft/min

KEY BOARD LCD Display

FdUp	FLOW => 99.9 %

ทำหน้าที่ในการเพิ่มอัตราการไหล

KEY BOARD LCD Display

FdDw	FLOW => 0.0 %

ทำหน้าที่ในการลดอัตราการไหล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
การออกแบบซอฟท์แวร์ เพื่อใช้ในการแสดงผล ของตัวจำลองกระบวนการทางความร้อน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. วิธีการออกแบบซอฟท์แวร์โปรแกรม PROGRAM SIMULATOR

1.1 แนวคิดการออกแบบโปรแกรม

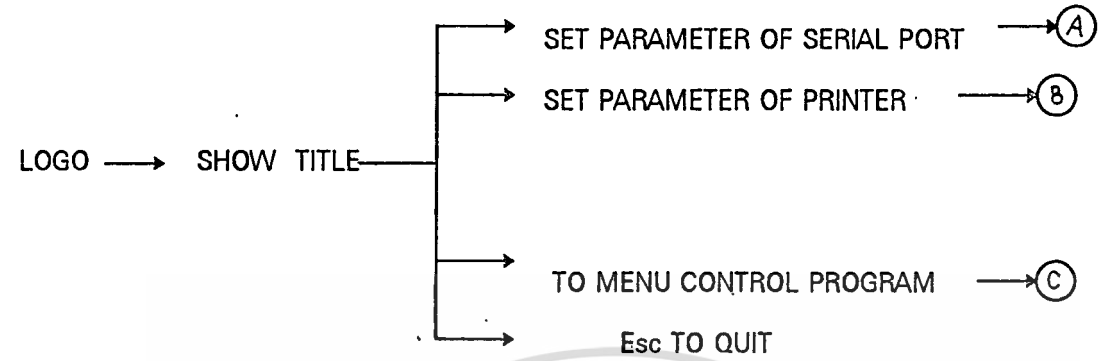
การแสดงผลของสัญญาณต่าง ๆ ของโปรแกรม PROCESS SIMULATOR เป็นการวัดค่าตัวแปรต่าง ๆ ที่อยู่ในกระบวนการ เช่น ตัวแปรการจัดการ (Manipulated Variable , MV) ,ตัวแปรโปรเซส (Process Variable , PV) ,อุณภูมิของเหลว (Temp Liquid),ปริมาณของเหลว (Feed Liquid),ปริมาณไอน้ำ (Feed Steam) และอุณหภูมิไอน้ำ (Temp Steam) โดยการรับสัญญาณเหล่านี้จากพอร์ตอนุกรม RS-232

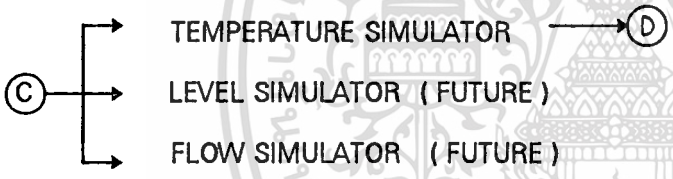
โปรแกรม PROCESS SIMULATOR เป็นโปรแกรมสำหรับแสดงผลการตอบสนองของพารามิเตอร์ต่างๆ ของตัวดิจิตอล คอนโทรลเลอร์ซึ่งมีการแสดงผลเป็นแบบกราฟบันทึกแนวโน้มข้อมูล (Line graph) ที่มีสีสันสวยงามง่ายต่อการเข้าใจ ในการออกแบบโปรแกรมจะต้องจัดการหัวข้อหลักดังนี้

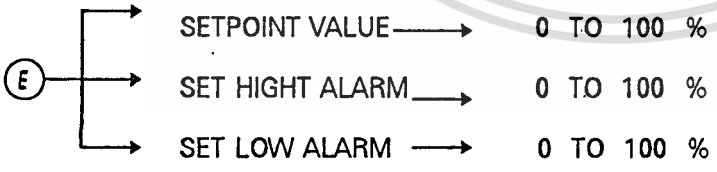
- สร้างเมนูเป็นแบบป๊อป-อัพพูลดาวน์เมนู (pop-up pull down menu) มีเมนูย่อย (sub menu) ให้เลือกใช้ โดยการเลื่อนแถบแสงซึ่งง่ายต่อการใช้งาน
- มีฟังก์ชั่นการทำงานเกี่ยวกับกราฟบันทึกแนวโน้มข้อมูล (Line graph) ซึ่งแสดงค่าของตัวแปร 6 ตัวต่าง ๆ ดังกล่าวข้างต้น โดยการเทียบสเกล 0 ถึง 100 เปอร์เซนต์
- มีการแสดงเส้นสัญญาณจุดที่ต้องการ (Set point), ขอบเขตบน (Hight alarm) และขอบเขตล่าง (Low alarm) โดยการป้อนค่าทางคีย์บอร์ด
- มีเมนูย่อยในส่วนของการให้เซทค่าพารามิเตอร์ต่าง ๆ ของพอร์ตอนุกรม (SET PARAMETER OF SERIAL PORT)
- มีเมนูย่อยในส่วนของการให้เซทค่าพารามิเตอร์ต่าง ๆ ของการพิมพ์ (SET PARAMETER OF PRINTER)
- ในโปรแกรมนี้มีการเลือกใช้การจำลองกระบวนการแบบอุณหภูมิ(TEMPERATURE SIMULATOR) และได้เตรียมสำรองในส่วนของการใช้งานการจำลองกระบวนการต่าง ๆ เช่น LEVEL SIMULATOR และ FLOW SIMULATOR เพื่อความสะดวกในอนาคตอีกด้วย
- มีการแสดงรูปบล็อก (BLOCK DIAGRAM) ของกระบวนการ

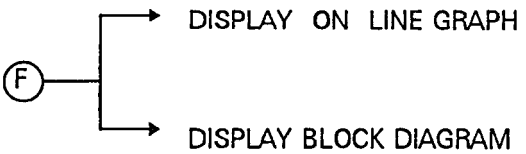
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.2 แผนผังการทำงานของเมนูซอฟท์แวร์โปรแกรม (particular of menu)









เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.3 เกี่ยวกับไฟล์เมนู (about operator of file menu)

ในการเขียนโปรแกรมเมนู ได้ใช้ภาษาซีเขียนขึ้นมา ซึ่ง PROJECT FILE ได้จากการ LINK EXE FILE ด้วย FILE นามสกุล OBJ ที่มาจาก source file ดังนี้คือ

PROJECT .PRJ

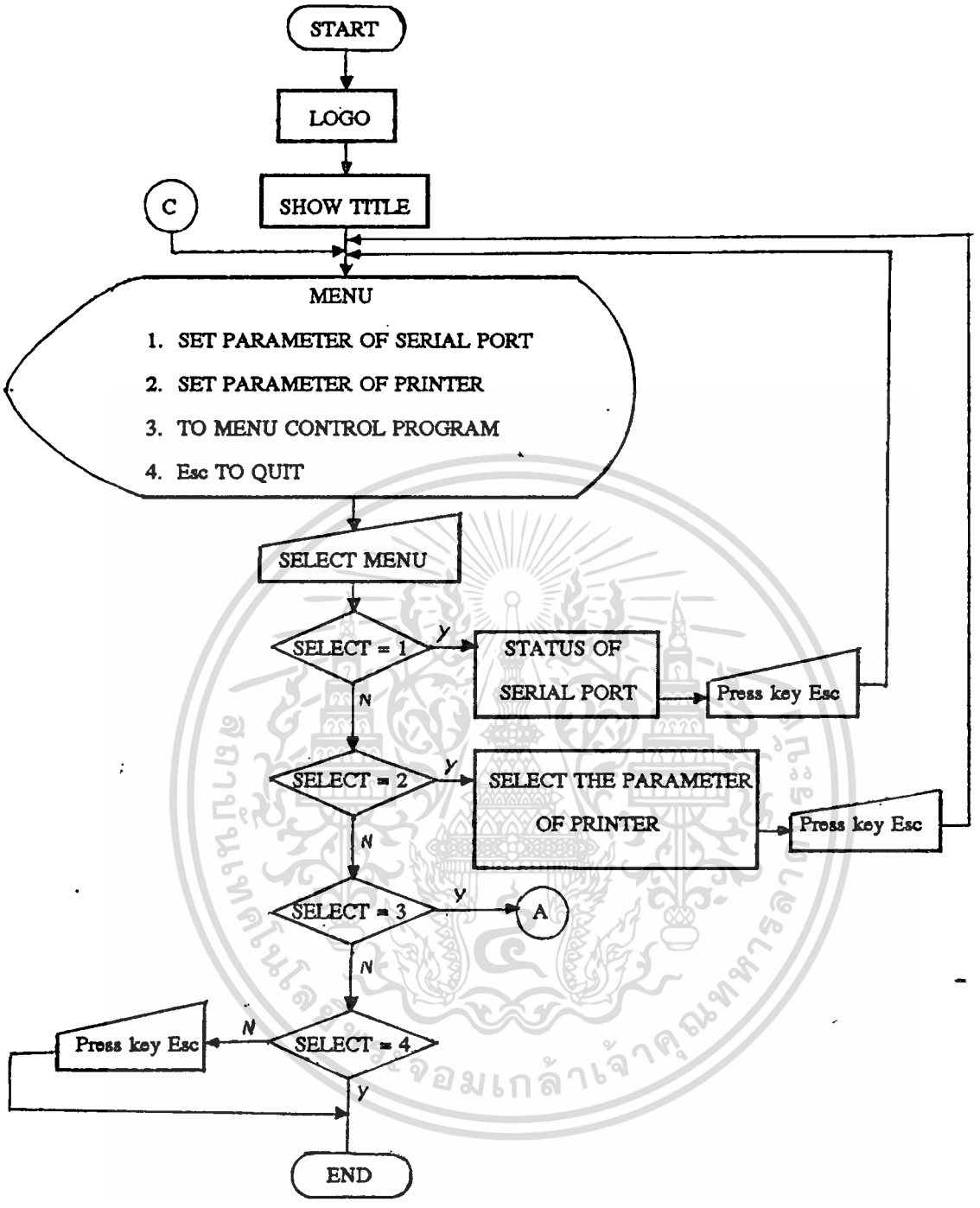
1. main .c เป็นโปรแกรมหลักในการทำงานที่จะเรียกใช้ FILE ต่าง ๆ
2. sub .c เป็นโปรแกรมสร้างเมนูตามที่ต้องการ
3. para .c เป็นโปรแกรมกำหนดพารามิเตอร์ SETPOINT ,HIGHT ALARM และ LOW ALARM
4. logo .c เป็นโปรแกรมแสดงการเริ่มต้นของโปรแกรมที่เรียกใช้
5. interf .c เป็นโปรแกรมในการกำหนด ตรวจสถานะ และเตรียมรับข้อมูลจากพอร์ตอนุกรม RS-232
6. port .c เป็นโปรแกรมกำหนดพอร์ต และ เครื่องพิมพ์
7. disp .c กำหนดการแสดงกราฟบันทึกแนวโน้มข้อมูล (Line graph) และ Block Diagram

2. โฟล์ชาร์ทการทำงานของโปรแกรม

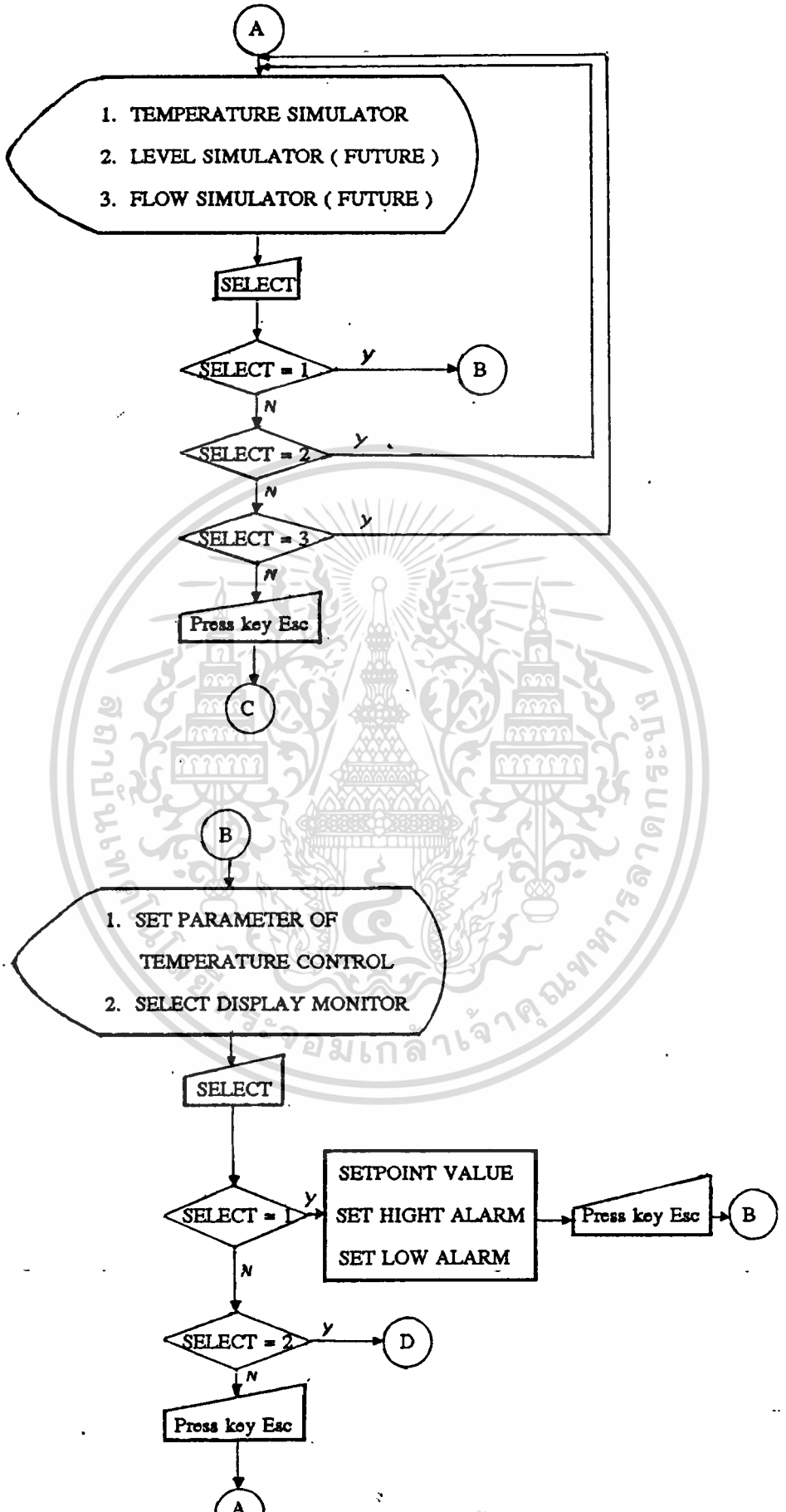
โปรแกรมจะทำงานด้วยการกดคีย์แต่ละคีย์ สำหรับเลือกฟังก์ชั่นที่ต้องการต่าง ๆ ของเมนู โดยคีย์ที่ออกแบบไว้มีหน้าที่ดังนี้

1. กดคีย์ Enter เป็นการเปลี่ยนไปหน้าจอถัดไป
2. กดคีย์ ลูกศรขึ้นลง↑ ↓ เป็นการเลื่อนแถบแสงเพื่อเลือกฟังก์ชั่นต่าง ๆ ตามต้องการ
3. กดคีย์ Esc เป็นการออกจากหน้าจอปัจจุบัน

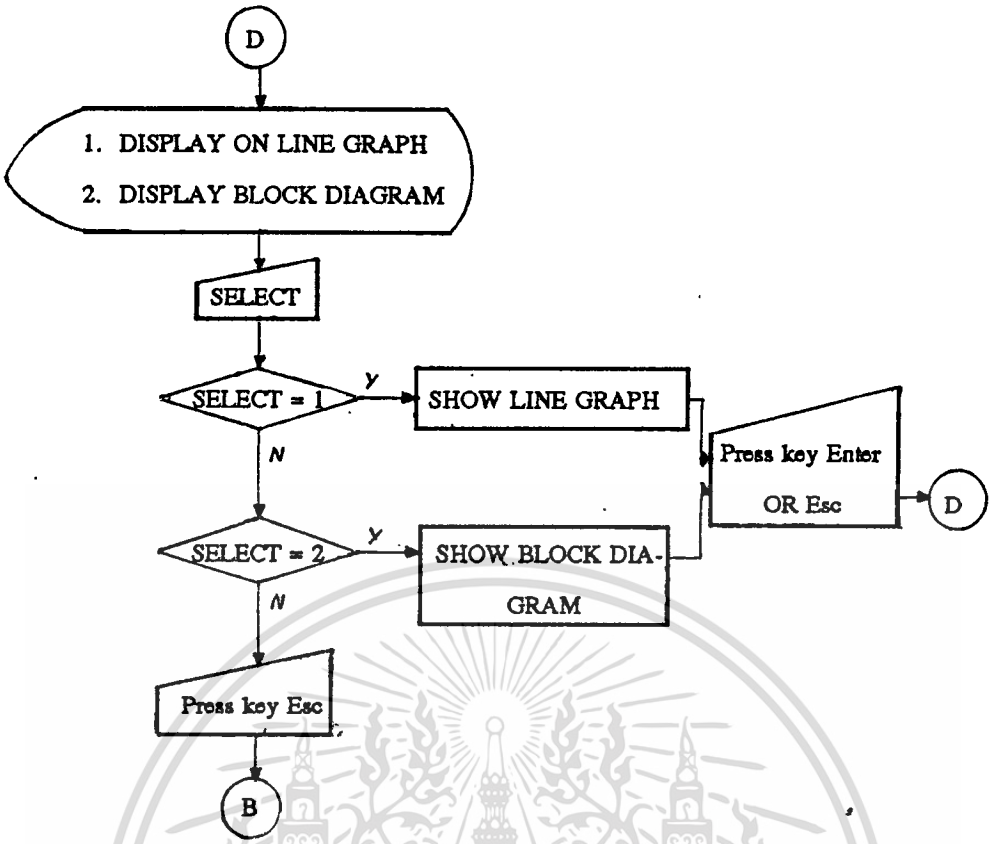
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



3. การใช้ภาษาซี เขียนโปรแกรมและปัญหาที่มักเกิดขึ้น

ภาษาซีเป็นภาษาชั้นสูงภาษาหนึ่งที่มีความคล่องตัวของภาษา และเหมาะแก่การใช้งานทั่ว ๆ ไป เป็นภาษาโครงสร้างที่มีฟังก์ชั่นใช้งานจำนวนมากให้เลือกใช้ ซอร์สโค้ดที่ได้จากภาษาซีมีขนาดเล็ก การประยุกต์ใช้งานจึงง่ายต่อการเรียนรู้และนำมาใช้ จึงได้นำจุดเด่นของภาษาซีมาใช้นั่นก็คือ การมีโครงสร้างของภาษาที่ดี

ภาษาซี นับเป็นภาษาที่อาจเรียกได้ว่าเป็นภาษามาตรฐานสำหรับการเขียนโปรแกรมบนระบบคอมพิวเตอร์ แม้ในการเขียนโปรแกรมที่เข้าถึงตัวระบบ ภาษาซีก็มีความอ่อนตัวถึงระดับที่ใกล้เคียงกับภาษาแอสเซมบลี ดังนั้นการเขียนโปรแกรมจึงค่อนข้างง่ายแต่ก็มีปัญหาสำหรับผู้ที่พึ่งจะศึกษาคือ ในส่วนของการอินเตอร์เฟสซึ่งจะต้องเขียนเพื่อใช้งานเฉพาะด้าน และยังอาจจะต้องเขียนอินคลูดไฟล์ (INCLUDE FILE) เพื่อเรียกใช้อีกด้วย จึงต้องศึกษาการติดต่อสื่อสารข้อมูลด้วยพอร์ตอนุกรม RS-232 ให้เข้าใจ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. การรับข้อมูลจากพอร์ตอนุกรม RS-232 ด้วยภาษาซี

การเรียกใช้งานพอร์ตอนุกรมสามารถทำได้ 3 วิธีคือ ผ่านทางดอส ,ผ่านทาง BIOS และการเขียนโปรแกรมควบคุมพอร์ตอนุกรมโดยตรง วิธีที่ง่ายที่สุดคือ การเรียกใช้ผ่าน BIOS อินเตอร์รัพต์หมายเลข 14 โดยจะต้องเตรียมสถานะเริ่มต้นของพอร์ต ,การตรวจสอบสถานะของพอร์ตอนุกรม และการรับข้อมูลผ่านทางพอร์ตอนุกรม

ในโครงงานนี้ ได้เตรียมสถานะพอร์ตอนุกรมไว้ดังนี้คือ ตั้งให้พอร์ตมีอัตรา 9600 boud ,ไม่มีพาริตี้ ,มี 1 บิตสิ้นสุด และใช้ 8 บิตต่อ 1 ไบต์ข้อมูล ซึ่งจะได้รหัสคือ 0xe3

ได้เขียนโปรแกรมรับข้อมูลผ่านทางพอร์ตอนุกรม โดยรับมาจากดิจิตอลคอนโทรลเลอร์ และนำมาแสดงในรูปของกราฟบันทึกแนวโน้มข้อมูล (Line graph) ซึ่งถือว่าเป็น real time คือรับข้อมูลแล้วมาแสดงทางจอเลย ข้อมูลที่แสดงไปแล้วจะไม่นำมาแสดงซ้ำอีก

โปรแกรมภาษาซีที่ได้เขียนไว้เพื่อนำมาใช้งานในโครงงานนี้ ทั้งในส่วนของการโชว์เมนู ,การแสดงผล และการอินเตอร์เฟสรับข้อมูลมาแสดงผลนั้น ดูได้จากภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
การทดลองและสรุปผลการทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

การทดลองและผลการทดลองกระบวนการด้วย PID

การทดลอง

สำหรับในบทนี้จะเป็นการนำกระบวนการจำลองมาทดลองใช้กับเครื่องควบคุม PID ซึ่งจะเป็นการควบคุมอุณหภูมิของระบบให้มีค่าอยู่ในจุด setpoint การทดลองนี้จะใช้อุปกรณ์ในการต่อร่วมดังนี้

1. เครื่องจำลองกระบวนการ
2. เครื่องควบคุม PID
3. คอมพิวเตอร์
4. สาย RS232

ลำดับขั้นการทดลอง

1. ทำการปรับเทียบ (Calibrate) อุปกรณ์ทุก ๆ หน่วย ให้ได้ค่าตามมาตราฐานที่กำหนดไว้
2. ต่ออุปกรณ์ต่าง ๆ ที่เกี่ยวข้องกับกระบวนการจำลอง โดยต่อกระบวนการจำลองเข้ากับเครื่องคอมพิวเตอร์โดยผ่าน RS232 สัญญาน PV จะต่อเข้ากับจุดต่อ 0 และ 2 ของเครื่องควบคุม และสัญญาน MV จะต่อเข้ากับจุดต่อ 5 และ 6 ของเครื่องควบคุม
3. เริ่มกระบวนการควบคุม โดยการโหลด (Load) โปรแกรมจาก MCS-51 เข้าเครื่องคอมพิวเตอร์ จาก File ชื่อ present3.bat จอคอมพิวเตอร์จะแสดงผลเป็นรูป MENU หลัก ทำการเลือกโหมดการควบคุมอุณหภูมิและจะเข้าสู่ Line graph
4. ปรับโหมดเครื่องควบคุมให้เป็นแบบ Manual ลดสัญญาน MV ให้เหลือ 0% ทำการตั้งค่าเป้าหมาย (SV) แล้วจึง RUN กระบวนการ เพิ่มค่า MV ให้มีค่าใกล้เคียงค่าเป้าหมาย
5. เปลี่ยนโหมดการควบคุมจาก Manual ไปเป็นแบบอัตโนมัติ (Automatic) 6. ทำการปรับค่า Proportional (P) ค่าของ Integral (I) และค่าของ Derivative(D) สังเกตุการเปลี่ยนแปลงของสัญญาน MV ว่าเข้าสู่จุดเป้าหมายหรือไม่ ถ้าไม่ได้ให้เปลี่ยนค่าของ P, I และ D จนกว่าจะสามารถควบคุมกระบวนการให้มีค่าตามที่ต้องการ
7. ทดลองเปลี่ยนค่า SV และสังเกตุว่าเครื่องควบคุมสามารถควบคุมกระบวนการให้มีค่าตามที่ต้องการหรือไม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วิธีการปรับตัวควบคุม

การปรับตัวควบคุมกระบวนการให้ได้ผลการควบคุมที่เหมาะสม กระทำได้โดยการปรับค่า GINT, INTERGRAL และ DERIVATIVE ให้เหมาะสม การกำหนดค่าต่าง ๆ เหล่านี้ แตกต่างกันไปตามคุณลักษณะของกระบวนการ ได้มีนักคณิตศาสตร์คิดวิธีที่จะหาทางตั้งค่า PID เพื่อให้ได้การควบคุมที่มีคุณภาพที่ดีที่สุดอยู่หลายวิธีด้วยกัน แต่ในการทดลองนี้จะใช้วิธีการปรับการควบคุมแบบลองถูกลองผิด (TRIAL AND ERROR)

วิธีการปรับตัวควบคุมโดยการใช้ TRIAL AND ERROR METHOD

วิธีการทั่ว ๆ ไปของ TRIAL AND ERROR METHOD คือการปรับ ACTION ของตัวควบคุมทุก ACTION ไว้กว้าง ๆ ก่อน แล้วค่อยๆ ลดลงให้แคบที่ละน้อย จนกระทั่งได้เสถียรภาพตามที่ต้องการ วิธีนี้จำเป็นต้องให้โหลดมีการเปลี่ยนแปลงเพื่อรบกวน ซึ่งทำได้โดยการปรับค่าเป้าหมายไปเพียงเล็กน้อย อาจจะเพิ่มหรือลดก็ได้ และจะต้องมากพอที่จะรบกวนระบบได้ และการเปลี่ยนค่าเป้าหมาย เพื่อปรับตัวควบคุมด้วยวิธีนี้ จะต้องให้เท่ากันทุกครั้ง

การปรับ PID มีขั้นตอนการทำดังนี้

1. ตัด INTEGRAL ACTION และ DERIVATIVE ACTION ออกไป โดยตั้งค่า INTEGRAL TIME (Ti) ไว้ที่ค่าสูงสุด และ DERIVATIVE TIME (Td) ไว้ที่ค่าต่ำสุด

2. เริ่มต้นการปรับ PROPORTIONAL BAND (PB) ไว้ที่ค่าสูงสุด แล้วค่อย ๆ ลด PB ลงมาที่ละน้อยเป็น STEP หลังจากการปรับ PB ไปแต่ละครั้งให้ลองเปลี่ยนค่าเป้าหมายเพื่อรบกวนระบบแล้วสังเกตุลักษณะของการควบคุมแล้วปรับ PB ใหม่ สลับกันไปเรื่อย ๆ จนกระทั่งได้ลักษณะการควบคุมที่ดี

3. ตั้งค่า PB ไว้ตามข้อ 2 แล้วค่อย ๆ ลดค่า Ti ลงมาทีละน้อย ๆ เป็น STEP สลับกัน การเปลี่ยนค่าเป้าหมายเพื่อรบกวนระบบจนกระทั่งเกิดการแกว่ง และเพิ่ม Ti ให้มีค่าสูงขึ้นจนการแกว่งหายไป

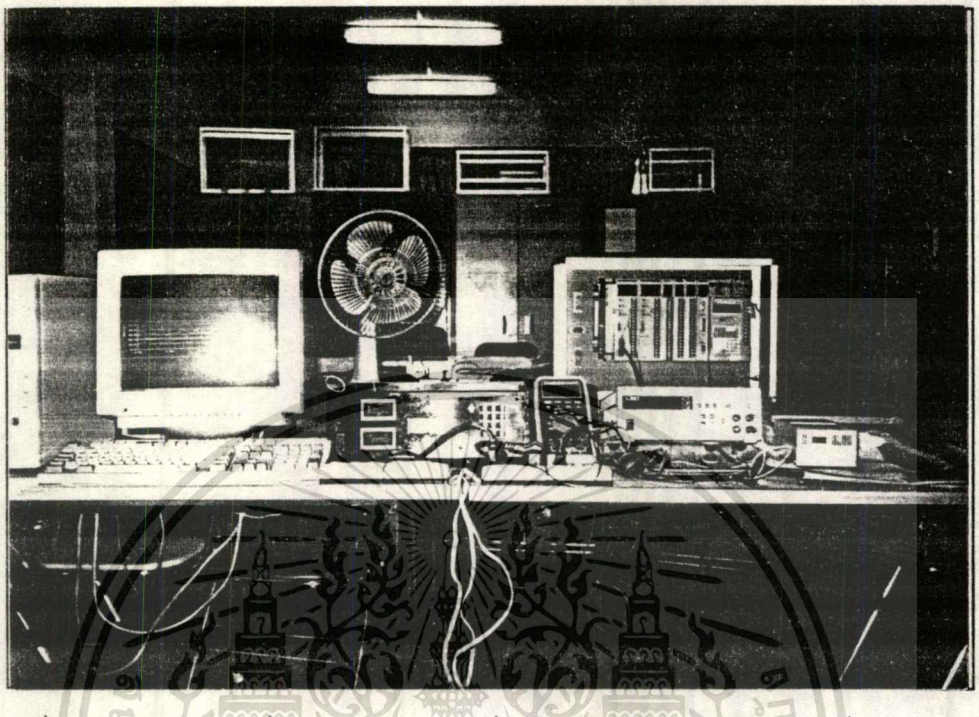
4. ตั้งค่า PB และ Ti ไว้ตามข้อ2 และข้อ 3 แล้วค่อย ๆ เพิ่มค่า Td ทีละน้อยๆ เป็นSTEP จนกระทั่งระบบมีความเสถียรภาพตามต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

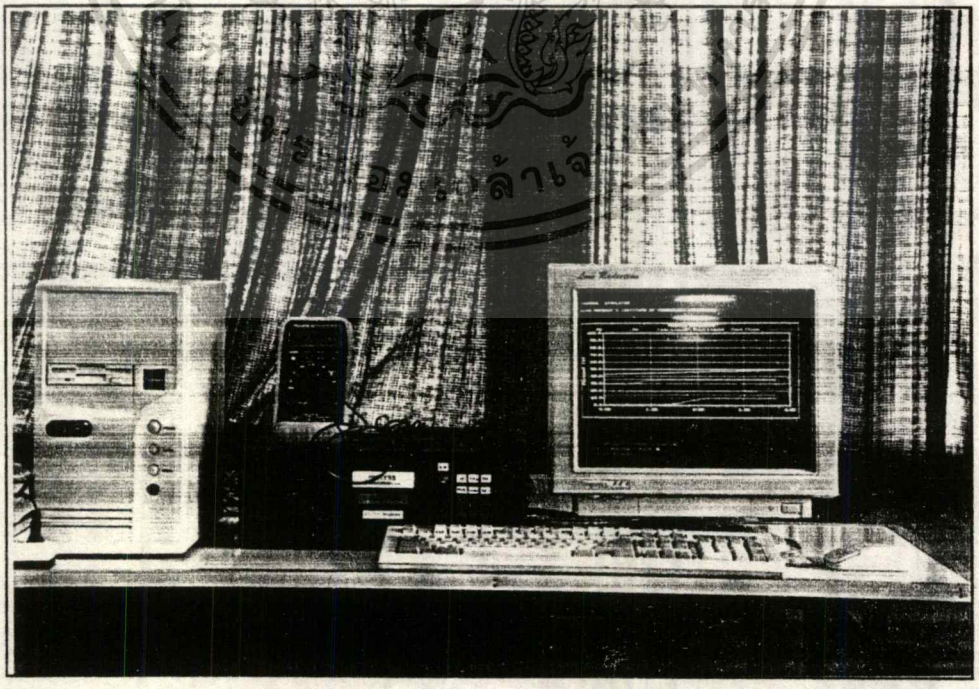
ผลการทดลอง

ผลการทดลองใช้เครื่องควบคุมแบบ PID ในการควบคุมกระบวนการจำลอง ปรากฏว่าสามารถทำการควบคุมได้เป็นอย่างดี โดยเมื่อตั้งค่าเป้าหมาย (SV) ไว้ที่ 30% และใช้วิธีการปรับ PID แบบ TRIAL AND ERROR METHOD โดยปรับค่า P ไว้ที่ 10% ค่าของ I ไว้ที่ 3400 Sec และค่าของ D ไว้ที่ 10 Sec ระบบกระบวนการจำลองมีความเสถียรภาพดีและเข้าใกล้ค่าเป้าหมาย เมื่อเพิ่มค่าเป้าหมายเป็น 40% ระบบก็ยังมีความเสถียรภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

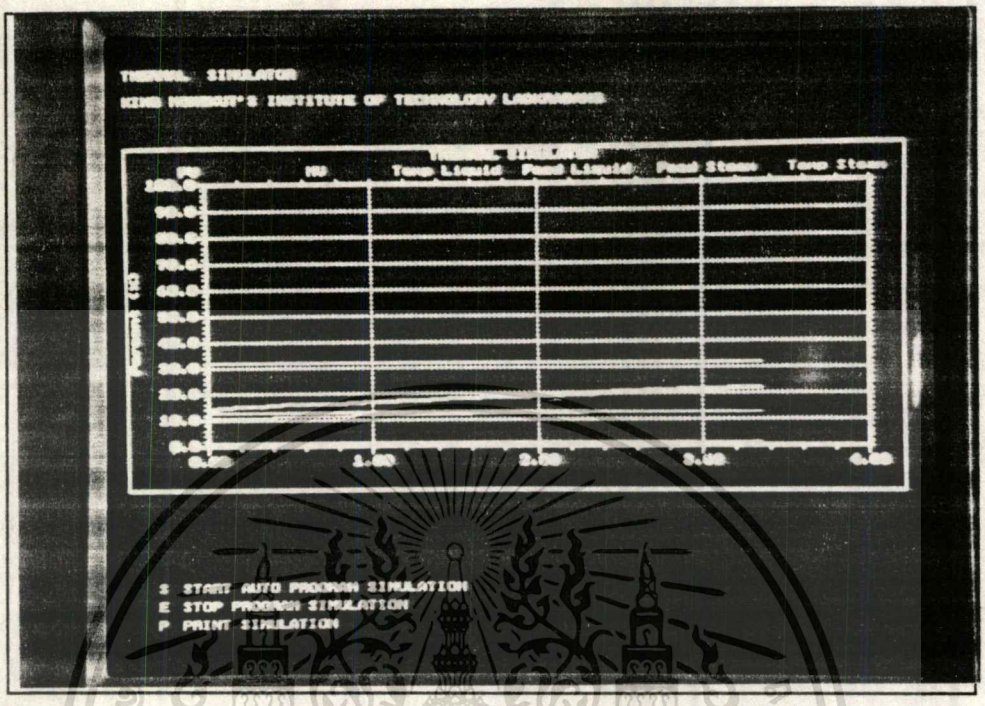


รูปที่ 5.1 ภาพแสดงเครื่องมือและอุปกรณ์ที่ใช้ในการต่อร่วมกันระหว่างเครื่องควบคุม กับ ตัวจำลองกระบวนการทางความร้อน

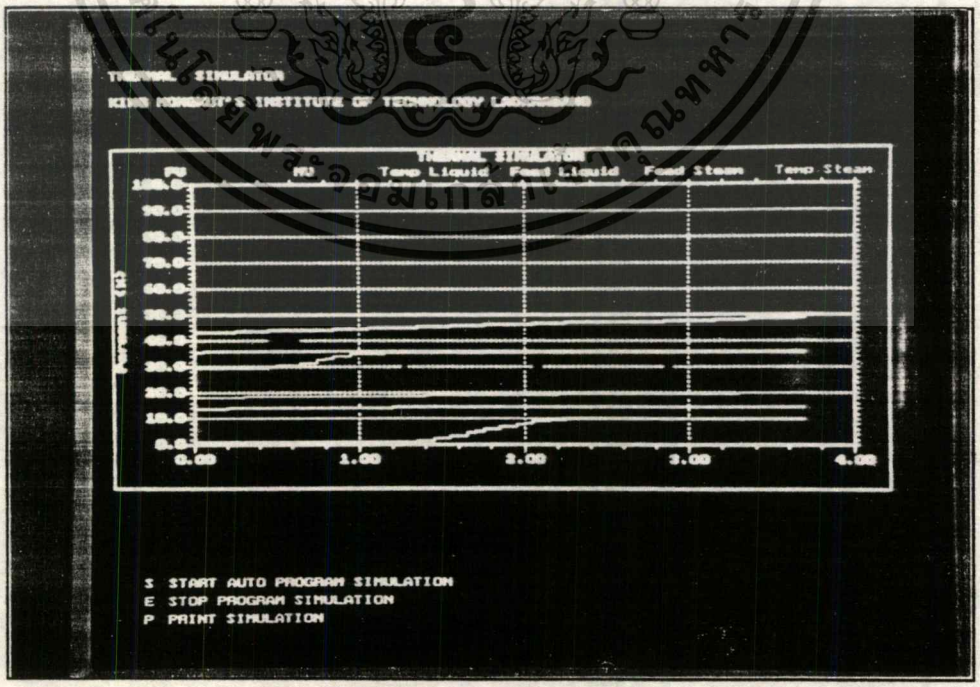


รูปที่ 5.2 แสดงการต่อร่วมกับคอมพิวเตอร์เพื่อแสดงผลของ ตัวจำลองกระบวนการทางความร้อนบนหน้าจอของคอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.3 แสดงผลการเปลี่ยนแปลงของสัญญาณควบคุม (MV)
และผลการเปลี่ยนแปลง ของสัญญาณทางด้านอุณหภูมิขาออก (PV)



รปที่ 5.4 แสดงการเปลี่ยนแปลงของสัญญาณรบกวนทางด้านอุณหภูมิของ
ของไหล (Ti) และอัตราการเปลี่ยแปลงของ ของไหล (Fd)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
;************************************************
;*       SOFTWARE FOR PROCESS SIMULATOR         *
;*       SIMULATE  OF MODEL-Thermal Process     *
;*       BY..... MCS - 51                       *
;*       Programer By  Mr.Thawan Worachin       *
;************************************************

$INCLUDE "DATAPRJK.ASM"
ORG     0A000H

COMMAND     EQU     0E080H    ; Read-Write Register
READBUSY    EQU     0E081H    ; Read BF(Busy Flag) and address
WRITEDATA   EQU     0E082H    ; Write character
READDATA    EQU     0E083H    ; Read Data from DD ram
PORTA       EQU     0E020H     ; 8255 PORT-A
PORTB       EQU     0E021H     ; 8255 PORT-B
PORTC       EQU     0E022H     ; 8255 PORT-C
PORT_CTR    EQU     0E023H     ; 8255 CONTROL PORT
SYSCALL     EQU     0030H     ; System call
IC_DAC      EQU     0E040H
IC_ADC      EQU     0E042H        ; port input of ADC0804
B_DSP       EQU     0E001H        ; port control LED flag
P_DSP       EQU     0E002H        ; port enable LED flag
DISBUF      EQU     9F00H         ; display buffer (7 segment)
BUF         EQU     9F07H         ; KEY buffer
;##############################
; Internal Frag
SGNFg       EQU  00H
SGNDFg      EQU  01H
MULCY       EQU  02H  ;MULTIPLY CARRY FRAG
SGNIFg      EQU  03H
NO_DIGit    EQU  08H
TOGGLEFG    EQU  05H
FLAG        EQU  06H  ; data to enable LED flag
SUMFg       EQU  1BH
BKSUMFg     EQU  1CH
PSW_3       EQU  0D3H
AD_DWFlag   EQU  20H
AD_UPFlag   EQU  21H
DW_TEMP     EQU  22H
UP_TEMP     EQU  23H
DW_FEED     EQU  24H
UP_FEED     EQU  25H
MV_Flag     EQU  26H
;##############################
;INTERNAL REGISTOR
TEMP        EQU    17H
FEED        EQU    16H
WS          EQU    15H
TS          EQU    14H
TO          EQU    13H
ADBUFF      EQU    12H
DABUFF      EQU    11H
KEYBUFF     EQU    10H
;##############################
;********************
;*   MAIN PROGRAM   *
;********************
START:      JMP   MAINPROGRAME
            ORG    START+23H
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น ไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
INT0_SERV:  CLR    EX0
            PUSH   ACC
            PUSH   B
            PUSH   DPL
            PUSH   DPH
            LCALL  SUB_INT0
            JNB    P3.2,$
            POP    DPH
            POP    DPL
            POP    B
            POP    ACC
            SETB   EX0
            RETI
;==============================
SUB_INT0:   NOP
            PUSH 07H
            PUSH 06H
            PUSH 05H
            PUSH 04H
            PUSH 03H
            PUSH 02H
            PUSH 01H
            PUSH 00H
            MOV    A,#26H
            LCALL  0030H
            CJNE   A,#0FFH,CHK0
            SJMP   OUT_INT0
CHK0:       CJNE   A,#01H,CHK1
            CLR    AD_DWFlag
            SJMP   OUT_INT0
CHK1:       CJNE   A,#05H,CHK2
            CLR    AD_UPFlag
            SJMP   OUT_INT0
CHK2:       CJNE   A,#02H,CHK3
            CLR    DW_TEMP
            SJMP   OUT_INT0
CHK3:       CJNE   A,#06H,CHK4
            CLR    UP_TEMP
            SJMP   OUT_INT0
CHK4:       CJNE   A,#03H,CHK5
            CLR    DW_FEED
            SJMP   OUT_INT0
CHK5:       CJNE   A,#07H,OUT_INT0
            CLR    UP_FEED
OUT_INT0:   NOP
            POP    00H
            POP    01H
            POP    02H
            POP    03H
            POP    04H
            POP    05H
            POP    06H
            POP    07H
            RET

;==================================
INT1_SERV:  CLR    EX1
            PUSH   B
            PUSH   DPL
            PUSH   DPH
            PUSH   ACC
            LCALL  RS232C
            MOV    DPTR,#0E0CCH
            SETB   ACC.7
            MOVX   @DPTR,A
            JNB    P3.3,$ ;WAIT TILL INT1 GO HIGH
            POP    ACC
            POP    DPH
            POP    DPL
            POP    B
            SETB   EX1
            CLR    FLAG
            RETI
;============================================================
;SET EXTERNAL INTRRUPT
MAINPROGRAME: NOP
ST1_INT0:   MOV    A,#29H          ; Set interrupt vector
            MOV    DPTR,#INT0_SERV
            MOV    B,#1            ;EXTERNAL INTERRUPT-0
            LCALL  SYSCALL
INT_1:      MOV    DPTR,#INT1_SERV ;FOR CLOCK
            MOV    A,#29H          ;SET_VECT
            MOV    B,#3            ;EXTERNAL INTERRUPT-1
            LCALL  0030H           ;SYSCALL
            ;MOV    A,#07H         ;CLRSCR
            ;LCALL  0030H          ;SYSCALL
            SETB   EA              ;GLOBAL INTERRUPT
            SETB   EX0             ;EXTERNAL INTERRUPT
            SETB   EX1
            CLR    PX0
            SETB   PX1
;=======================================
;SET Mode RS232C
;*******************
;    TX_RS232C     *
;*******************
            MOV    TMOD,#21H
            MOV    SCON,#52H
            MOV    TH1,#0FDH
            SETB   TR1
;====================================
Init_DS1287: MOV   DPTR,#0E0CAH
             MOV   A,#00101111B
             MOVX  @DPTR,A
             MOV   DPTR,#0E0CBH
             MOV   A,#01010110B
             MOVX  @DPTR,A
             MOV   DPTR,#0E0CCH
             MOV   A,#11010000B
             MOVX  @DPTR,A
;====================================
;INITIAL 8255
             MOV   DPTR,#PORT_CTR
             MOV   A,#10011011B
             MOVX  @DPTR,A
             CLR   ACC
;====================================
; CLEAR DATA BUFFER
INITIAL:     MOV    R2,#20H
             MOV    DPTR,#9000H
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น มีทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
JC_M:       MOV     A,#00H
            MOVX    @DPTR,A
            INC     DPL
            DJNZ    R2,JC_M
            MOV     ADBUFF,#00H
            MOV     DABUFF,#00H
            MOV     TEMP,#00H     ;TEMP FEED
            MOV     FEED,#00H     ;FEED LIQUID
            ;---------------------------------------
            LCALL   INITLCD       ;SHOW TITLE LCD
            MOV     DPTR,#TITLE1
            LCALL   DISP1
            MOV     A,#40H        ;NEW ADDRESS
            LCALL   GOTOXY
            MOV     DPTR,#TITLE2  ;SHOW
            LCALL   DISP0
;=================================================
; INPUT VARIABLE
            ;====================
ST_LOOP:     NOP
             SETB    AD_UPFlag
             SETB    AD_DWFlag
             SETB    UP_TEMP
             SETB    DW_TEMP
             SETB    UP_FEED
             SETB    DW_FEED
             CLR     C
;=============================================
; CHECK AUTO/MANUAL
LOOP_8255:  MOV     DPTR,#PORTA
            MOVX    A,@DPTR
            ANL     A,#00000001B
            SETB    MV_Flag
            JNZ     MANUAL_M   ;IF A = 1 JUMP MANUAL
            LCALL   WriteAD    ;Present  Input
            LCALL   INIT_1
            SJMP    OUT_MA
;=============================================
MANUAL_M:   CLR     MV_Flag
            LCALL   INIT_1     ;Previous Input
;=============================================
OUT_MA:     NOP
; BLOCK_1
        MOV     R5,#65H
        MOV     R6,#64H
        MOV     R7,#00H
        MOV     R4,#00H
        MOV     A,ADBuff
        LCALL   MULT16_16C     ;[R6:R5 * R4:A] = [R2:R1:R0]
        MOV     R7,#08H        ;NO_DIGIT
        LCALL   SHIFTDIG
        LCALL   OVERLIMIT2     ;O/P-->R1:R0
;---------------------------
; SAVE PREVIOUS VARIABLE
;---------------------------
        MOV     DPTR,#9010H
        MOV     A,R0
        MOVX    @DPTR,A
        INC     DPL
        MOV     A,R1
        MOVX    @DPTR,A
;=============================
; BLOCK_2
        MOV     R5,#4CH
        MOV     R6,#00H
        MOV     R7,#00H
        MOV     R4,01H
        MOV     A,00H
        LCALL   MULT16_16C
        MOV     R7,#08H
        LCALL   SHIFTDIG
        LCALL   OVERLIMIT2
;=============================
; BLOCK_3
        MOV     R2,#00H
        MOV     DPTR,#9001H
        MOVX    A,@DPTR
        MOV     R7,A
        DEC     DPL
        MOVX    A,@DPTR
        LCALL   ADD24_16C
        LCALL   Overlimit2
        LCALL   OverWS
        MOV     WS,R1
        ;-------------------------
        ; SAVE PREVIOUS VARIABLE
        ;-------------------------
        MOV     DPTR,#9012H
        MOV     A,R0
        MOVX    @DPTR,A
        INC     DPL
        MOV     A,R1
        MOVX    @DPTR,A
;=============================
; BLOCK_4
        MOV     R5,00H      ;<--R0
        MOV     R6,01H      ;<--R1
        MOV     R7,#00H
        MOV     R4,#00H
        MOV     A,#0A2H
        LCALL   MULT16_16C
        MOV     R7,#08H
        LCALL   SHIFTDIG
        LCALL   OVERLIMIT2
;=============================
; BLOCK_5
        MOV     R2,#00H
        MOV     DPTR,#9003H ;
        MOVX    A,@DPTR
        MOV     R7,A
        DEC     DPL
        MOVX    A,@DPTR
        LCALL   ADD24_16C   ;
        LCALL   Overlimit2  ;
;=============================
; BLOCK_6
        MOV     R2,#00H
        MOV     R7,#00H
        MOV     A,#34H
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        LCALL   ADD24_16C    ;[R2:R1:R0 + R7:A]
        LCALL   Overlimit2   ;-->R1:R0
;================================
; BLOCK_7
        MOV     R5,00H       ;<--R0
        MOV     R6,01H       ;<--R1
        MOV     R7,#00H
        MOV     R4,#00H
        MOV     A,#24H
        LCALL   MULT16_16C   ;[R6:R5 * R4:A]
        MOV     R7,#08H      ;NO_DIGIT
        LCALL   SHIFTDIG
        LCALL   OVERLIMIT2   ;O/P-->R1:R0
;================================
; BLOCK_8
        MOV     R2,#00H
        MOV     DPTR,#9005H ;
        MOVX    A,@DPTR
        MOV     R7,A
        DEC     DPL
        MOVX    A,@DPTR
        LCALL   ADD24_16C    ;[R2:R1:R0 + R7:A]
        LCALL   Overlimit2   ;-->R1:R0
        MOV     TS,R1
;----------------------------
; SAVE PREVIOUS VARIABLE
;----------------------------
        MOV     DPTR,#9016H
        MOV     A,R0
        MOVX    @DPTR,A
        INC     DPL
        MOV     A,R1
        MOVX    @DPTR,A
;============================================
; BLOCK_9
        MOV     R5,00H       ;<--R0
        MOV     R6,01H       ;<--R1
        MOV     R7,#00H
        MOV     R4,#01H
        MOV     A,#0ECH
        LCALL   MULT16_16C   ;[R6:R5 * R4:A]
        MOV     R7,#08H      ;NO_DIGIT
        LCALL   SHIFTDIG
        LCALL   OVERLIMIT2   ;O/P-->R1:R0
;============================================
; BLOCK_10
        MOV     R2,#00H
        MOV     DPTR,#9007H ;
        MOVX    A,@DPTR
        MOV     R7,A
        DEC     DPL
        MOVX    A,@DPTR
        LCALL   ADD24_16C    ;[R2:R1:R0 + R7:A]
        LCALL   Overlimit2   ;-->R1:R0
;============================================
; BLOCK_11
        CLR     C
        MOV     R2,#00H
        MOV     DPTR,#9009H ;
        MOVX    A,@DPTR
        MOV     R7,A
        DEC     DPL
        MOVX    A,@DPTR
        MOV     R6,A
        LCALL   SUB16_16C    ;[R1:R0 - R7:R6] = :R1
        JC      Under_16
        LCALL   Overlimit2   ;-->R1:R0
        SJMP    OutB_16
Under_16:   MOV     R0,#00H
            MOV     R1,#00H
OutB_16:    NOP              ;-->R1:R0
;============================
; BLOCK_12
        MOV     R5,00H       ;<--R0
        MOV     R6,01H       ;<--R1
        MOV     R7,#00H
        MOV     R4,#00H
        MOV     A,#05H
        LCALL   MULT16_16C   ;[R6:R5 * R4:A]
        MOV     R7,#08H      ;NO_DIGIT
        LCALL   SHIFTDIG
        LCALL   OVERLIMIT2   ;O/P-->R1:R0
;============================================
; BLOCK_13
        MOV     R2,#00H
        MOV     DPTR,#900BH ;
        MOVX    A,@DPTR
        MOV     R7,A
        DEC     DPL
        MOVX    A,@DPTR
        LCALL   ADD24_16C    ;[R2:R1:R0 + R7:A]
        LCALL   Overlimit2   ;-->R1:R0
        MOV     TO,R1
        MOV     DABuff,R1
        ;LCALL  OUTPUT        ; OUTPUT TO DAC0832
;----------------------------
; SAVE PREVIOUS VARIABLE
;----------------------------
        MOV     DPTR,#9014H
        MOV     A,R0
        MOVX    @DPTR,A
        INC     DPL
        MOV     A,R1
        MOVX    @DPTR,A
        SETB    FLAG
JJJJ:   LCALL   OUTPUT
        JB      FLAG,JJJJ
        SETB    FLAG
KKKK:   LCALL   MANUAL
OUTKEY: JB      FLAG,KKKK
OUTLOOP:LJMP ST_LOOP
;============================================
RS232C:         NOP
                PUSH 07H
                PUSH 06H
                PUSH 05H
                PUSH 04H
                PUSH 03H
                PUSH 02H
                PUSH 01H
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น ห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        MOV     A,R0
        MOVX    @DPTR,A
        INC     DPL
        MOV     A,R1
        MOVX    @DPTR,A
; A3 = 00:DD * <S7>  ====> [R6:R5 * R4:A]
        MOV     R5,#0DEH
        MOV     R6,#00H
        MOV     R7,#00H
        MOV     DPTR,#9017H     ;S11
        MOVX    A,@DPTR
        MOV     R4,A
        DEC     DPL
        MOVX    A,@DPTR
        LCALL   MULT16_16C      ;
        MOV     R3,#00H
        MOV     R7,#08H         ;NO_DIGIT
        LCALL   SHIFTDIG
        LCALL   OVERLIMIT2      ;O/P
        MOV     DPTR,#9004H
        MOV     A,R0
        MOVX    @DPTR,A
        INC     DPL
        MOV     A,R1
        MOVX    @DPTR,A
; A4 = 00:9E * TEMP  ====> [R6:R5 * R4:A]
        MOV     R5,#09FH
        MOV     R6,#00H
        MOV     R7,#00H
        MOV     R4,#00H
        MOV     A,TEMP
        LCALL   MULT16_16C
        MOV     R3,#00H
        MOV     R7,#08H         ;NO_DIGIT
        LCALL   SHIFTDIG
        LCALL   OVERLIMIT2      ;O/P
        MOV     DPTR,#9006H
        MOV     A,R0
        MOVX    @DPTR,A
        INC     DPL
        MOV     A,R1
        MOVX    @DPTR,A
; A5 = 00:35 * FEED  ====> [R6:R5 * R4:A]
        MOV     R5,#00H
        MOV     R6,#36H
        MOV     R7,#00H
        MOV     R4,#00H
        MOV     A,FEED
        LCALL   MULT16_16C
        MOV     R3,#00H
        MOV     R7,#08H         ;NO_DIGIT
        LCALL   SHIFTDIG
        LCALL   OVERLIMIT2      ;O/P
; A5/1
        MOV     R2,#00H
        MOV     R7,#00H
        MOV     A,#0B2H
        LCALL   ADD24_16C       ;
        LCALL   Overlimit2      ;-->R1:R0
        MOV     DPTR,#9008H
        PUSH 00H
        MOV     A,ADBUFF
        LCALL   SBYTE
        MOV     A,TEMP
        LCALL   SBYTE
        MOV     A,FEED
        LCALL   SBYTE
        MOV     A,WS
        LCALL   SBYTE
        MOV     A,TS
        LCALL   SBYTE
        MOV     A,TO
        LCALL   SBYTE
        POP   00H
        POP   01H
        POP   02H
        POP   03H
        POP   04H
        POP   05H
        POP   06H
        POP   07H
        RET
;=================================================
; PREVIOUS VARIABLE
INIT_1:
; A1 = 00:B5 * <S4>  ====> [R6:R5 * R4:A] => [R2:R1:R0]
        MOV     R5,#0B6H
        MOV     R6,#00H
        MOV     R7,#00H
        MOV     DPTR,#9013H
        MOVX    A,@DPTR
        MOV     R4,A
        DEC     DPL
        MOVX    A,@DPTR
        LCALL   MULT16_16C      ;-->R2:R1:R0
        MOV     R3,#00H
        MOV     R7,#08H         ;NO_DIGIT
        LCALL   SHIFTDIG
        LCALL   OVERLIMIT2      ;O/P-->R1:R0
        MOV     DPTR,#9000H
        MOV     A,R0
        MOVX    @DPTR,A
        INC     DPL
        MOV     A,R1
        MOVX    @DPTR,A
; A2 = 00:33 * <S10>  ====> [R6:R5 * R4:A] => [R2:R1:R0]
        MOV     R5,#34H
        MOV     R6,#00H
        MOV     R7,#00H
        MOV     DPTR,#9015H     ;S11
        MOVX    A,@DPTR
        MOV     R4,A
        DEC     DPL
        MOVX    A,@DPTR
        LCALL   MULT16_16C      ;-->R2:R1:R0
        MOV     R3,#00H
        MOV     R7,#08H         ;NO_DIGIT
        LCALL   SHIFTDIG
        LCALL   OVERLIMIT2      ;O/P-->R1:R0
        MOV     DPTR,#9002H
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        MOV     A,R0
        MOVX    @DPTR,A
        INC     DPL
        MOV     A,R1
        MOVX    @DPTR,A
; A6 = 00:FC * <S10>  ====> [R6:R5 * R4:A]
        MOV     R5,#0FDH
        MOV     R6,#00H
        MOV     R7,#00H
        MOV     DPTR,#9015H
        MOVX    A,@DPTR
        MOV     R4,A
        DEC     DPL
        MOVX    A,@DPTR
        LCALL   MULT16_16C
        MOV     R3,#00H
        MOV     R7,#08H         ;NO_DIGIT
        LCALL   SHIFTDIG
        LCALL   OVERLIMIT2      ;O/P
        MOV     DPTR,#900AH
        MOV     A,R0
        MOVX    @DPTR,A
        INC     DPL
        MOV     A,R1
        MOVX    @DPTR,A
; A7 = 00:E6 * <S13>  ====> [R6:R5 * R4:A]
        MOV     R5,#0E7H
        MOV     R6,#00H
        MOV     R7,#00H
        MOV     DPTR,#901BH
        MOVX    A,@DPTR
        MOV     R4,A
        DEC     DPL
        MOVX    A,@DPTR
        LCALL   MULT16_16C      ;-->R2:R1:R0
        MOV     R3,#00H
        MOV     R7,#08H         ;NO_DIGIT
        LCALL   SHIFTDIG
        LCALL   OVERLIMIT2      ;O/P-->R1:R0
        MOV     DPTR,#900CH
        MOV     A,R0
        MOVX    @DPTR,A
        INC     DPL
        MOV     A,R1
        MOVX    @DPTR,A
        RET
;*****************************************
;SUB FOR INTERRUPT KEY
MANUAL:         PUSH ACC
                PUSH 07H
                PUSH 06H
                PUSH 05H
                PUSH 04H
                PUSH 03H
                PUSH 02H
                PUSH 01H
                PUSH 00H
;=========================================
ADUPKy:         JB      AD_UPFlag,ADDWKy
                MOV     A,ADBUFF
                CJNE    A,#0FFH,UPKY_N1
                MOV     ADBUFF,#0FFH
                LCALL   MVst
                LJMP    JP_OUTCHK
UPKY_N1:        JB      MV_Flag,DSP_MV1
                INC     ACC
DSP_MV1:        MOV     ADBUFF,A
                LCALL   MVst
                SETB    AD_UPFlag
                SETB    MV_Flag
                LJMP    JP_OUTCHK
ADDWKY:         JB      AD_DWFlag,UPKY_TEMP
                MOV     A,ADBUFF
                CJNE    A,#00H,DWKY_N2
                MOV     ADBUFF,#00H
                LCALL   MVst
                LJMP    JP_OUTCHK
DWKY_N2:        JB      MV_Flag,DSP_MV2
                DEC     ACC
DSP_MV2:        MOV     ADBUFF,A
                LCALL   MVst
                SETB    AD_DWFlag
                SETB    MV_Flag
                LJMP    JP_OUTCHK
;=============================================
UPKY_TEMP:      JB      UP_TEMP,DWKY_TEMP
                MOV     A,TEMP
                CJNE    A,#0FFH,UPKY_N3
                MOV     TEMP,#0FFH
                LCALL   TIst                ;<=======
                LJMP    JP_OUTCHK
UPKY_N3:        INC     ACC
                MOV     TEMP,A
                LCALL   TIst                ;<=======
                SETB    UP_TEMP
                LJMP    JP_OUTCHK
DWKY_TEMP:      JB      DW_TEMP,UPKY_FEED
                MOV     A,TEMP
                CJNE    A,#00H,DWKY_N4
                MOV     TEMP,#00H
                LCALL   TIst                ;<======
                LJMP    JP_OUTCHK
DWKY_N4:        DEC     ACC
                MOV     TEMP,A
                LCALL   TIst                ;<======
                SETB    DW_TEMP
                LJMP    JP_OUTCHK
;=============================================
UPKY_FEED:      JB      UP_FEED,DWKY_FEED
                MOV     A,FEED
                CJNE    A,#0FFH,UPKY_N5
                MOV     FEED,#0FFH
                LCALL   FSst
                SJMP    JP_OUTCHK
UPKY_N5:        INC     ACC
                MOV     FEED,A
                LCALL   FSst
                SETB    UP_FEED
                SJMP    JP_OUTCHK
DWKY_FEED:      JB      DW_FEED,JP_OUTCHK
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                MOV     A,FEED
                CJNE    A,#00H,DWKY_N6
                MOV     FEED,#00H
                LCALL   FSst
                SJMP    JP_OUTCHK
DWKY_N6:        DEC     ACC
                MOV     FEED,A
                LCALL   FSst
                SETB    DW_FEED
JP_OUTCHK:      NOP
                POP   00H
                POP   01H
                POP   02H
                POP   03H
                POP   04H
                POP   05H
                POP   06H
                POP   07H
                POP   ACC
                RET
;=============================================
; TEMP for feed Liquid
;=============================================
TIst:           PUSH ACC
                PUSH 07H
                PUSH 06H
                PUSH 05H
                PUSH 04H
                PUSH 03H
                PUSH 02H
                PUSH 01H
                PUSH 00H
;-------------------------------------
          MOV     DPTR,#COMMAND
          MOV     A,#1              ;CLEAR DISP
          MOVX    @DPTR,A
          LCALL   WAITBF
          MOV     A,TEMP
          MOV     DPTR,#TableDSP2 ;INDEX LSB
          MOVC    A,@A+DPTR
          MOV     R3,A              ;SAVE DATA
          MOV     DPTR,#TableDSP3 ;INDEX MSB
          MOV     A,TEMP            ;R7
          MOVC    A,@A+DPTR
          MOV     R4,A              ;SAVE DATA
          PUSH    03H
          PUSH    04H
          ;-------------------------
          MOV     DPTR,#TI_TEMP
          MOV     R0,#08H
STR100:   MOVX    A,@DPTR
          LCALL   WRITE_CHR
          INC     DPTR
          DJNZ    R0,STR100
          ;-------------------------------------
          MOV     A,#40H            ;New Address
          MOV     DPTR,#COMMAND
          SETB    ACC.7             ;SET DDRAM INST
          MOVX    @DPTR,A
          LCALL   WAITBF
          ;-------------------------------------
          POP     04H
          POP     03H
          MOV     DPH,R4
          MOV     DPL,R3
          MOV     R0,#08H
STR200:   MOVX    A,@DPTR
          LCALL   WRITE_CHR
          INC     DPTR
          DJNZ    R0,STR200
          POP   00H
          POP   01H
          POP   02H
          POP   03H
          POP   04H
          POP   05H
          POP   06H
          POP   07H
          POP   ACC
          RET
;=============================================
; FEED_FLOW for feed Liquid
;=============================================
FSst:           PUSH ACC
                PUSH 07H
                PUSH 06H
                PUSH 05H
                PUSH 04H
                PUSH 03H
                PUSH 02H
                PUSH 01H
                PUSH 00H
;-------------------------------------
          MOV     DPTR,#COMMAND
          MOV     A,#1              ;CLEAR DISP
          MOVX    @DPTR,A
          LCALL   WAITBF
          MOV     A,FEED
          MOV     DPTR,#TableDSP2 ;INDEX LSB
          MOVC    A,@A+DPTR
          MOV     R3,A              ;SAVE DATA
          MOV     DPTR,#TableDSP3 ;INDEX MSB
          MOV     A,FEED            ;R7
          MOVC    A,@A+DPTR
          MOV     R4,A              ;SAVE DATA
          PUSH    03H
          PUSH    04H
          ;-------------------------
          MOV     DPTR,#FS_FLOW
          MOV     R0,#08H
STR101:   MOVX    A,@DPTR
          LCALL   WRITE_CHR
          INC     DPTR
          DJNZ    R0,STR101
          ;-------------------------------------
          MOV     A,#40H            ;New Address
          MOV     DPTR,#COMMAND
          SETB    ACC.7             ;SET DDRAM
          MOVX    @DPTR,A
          LCALL   WAITBF
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น ทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        ;---------------------------------------
        POP     04H
        POP     03H
        MOV     DPH,R4
        MOV     DPL,R3
        MOV     R0,#08H
STR201: MOVX    A,@DPTR
        LCALL   WRITE_CHR
        INC     DPTR
        DJNZ    R0,STR201
        POP   00H
        POP   01H
        POP   02H
        POP   03H
        POP   04H
        POP   05H
        POP   06H
        POP   07H
        POP   ACC
        RET
;============================
; MANUAL VALVE for ADBUFF
;============================
MVst:       PUSH ACC
            PUSH 07H
            PUSH 06H
            PUSH 05H
            PUSH 04H
            PUSH 03H
            PUSH 02H
            PUSH 01H
            PUSH 00H
;-----------------------------------
        MOV     DPTR,#COMMAND
        MOV     A,#1            ;CLEAR DISP & HOME
        MOVX    @DPTR,A
        LCALL   WAITBF
        MOV     A,ADBUFF
        MOV     DPTR,#TableDSP2 ;INDEX LSB
        MOVC    A,@A+DPTR
        MOV     R3,A            ;SAVE DATA
        MOV     DPTR,#TableDSP3 ;INDEX MSB
        MOV     A,ADBUFF          ;R7
        MOVC    A,@A+DPTR
        MOV     R4,A            ;SAVE DATA
        PUSH    03H
        PUSH    04H
        ;-----------------------
        MOV     DPTR,#SP_PER
        MOV     R0,#08H
STR102: MOVX    A,@DPTR
        LCALL   WRITE_CHR
        INC     DPTR
        DJNZ    R0,STR102
        ;---------------------------------------
        MOV     A,#40H          ;New Address
        MOV     DPTR,#COMMAND
        SETB    ACC.7           ;SET DDRAM INST.
        MOVX    @DPTR,A
        LCALL   WAITBF
        ;---------------------------
        POP     04H
        POP     03H
        MOV     DPH,R4
        MOV     DPL,R3
        MOV     R0,#08H
STR202: MOVX    A,@DPTR
        LCALL   WRITE_CHR
        INC     DPTR
        DJNZ    R0,STR202
        POP   00H
        POP   01H
        POP   02H
        POP   03H
        POP   04H
        POP   05H
        POP   06H
        POP   07H
        POP   ACC
        RET
;============================
; Limit WS
OverWS: CLR   C
        MOV   A,#34H
        SUBB  A,R1
        JNC   OutWS
        MOV   R0,#0FFH
        MOV   R1,#33H
        MOV   R2,#00H
OutWS:  RET
;##################################
;INPUT  R0:R1 + A
;OUTPUT R0:R1
;R0=LSB
;R1=MSB
;A=ADDER
ADD16_8:  PUSH PSW
          CLR  C
          ADD  A,R0
          MOV  R0,A
          MOV  A,#00H
          ADDC A,R1
          MOV  R1,A
          POP  PSW
          RET
;####################################
;INPUT  R0*1,R1*10 + R7*10,A*1
;OUTPUT R7:A
ADD16_16: PUSH PSW
          PUSH 02H
          PUSH 03H
          MOV  03H,R7 ;KEEP
          MOV  R2,A   ;KEEP
          CLR  C
          ADD  A,R0
          MOV  R0,A
          MOV  A,R7
          ADDC A,R1
          MOV  R1,A
          MOV  07H,R3 ;GET
```

เอกสารนี้เป็นเอกสารสงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        MOV  A,R2   ;GET
        POP  03H
        POP  02H
        POP  PSW
        RET
;#####################################
; R1,R0 + R6 = R1,R0 -> C
ADD16_8C:  CLR  C
        MOV  A,R6
        ADD  A,R0
        MOV  R0,A
        MOV  A,#00H
        ADDC A,R1
        MOV  R1,A
        RET
;#####################################
;INPUT  R0*1,R1*10 + R7*10,A*1
;OUTPUT R7,A
ADD16_16C: PUSH 02H
        PUSH 03H
        MOV  03H,R7
        MOV  R2,A
        CLR  C
        ADD  A,R0
        MOV  R0,A
        MOV  A,R7
        ADDC A,R1
        MOV  R1,A
        MOV  07H,R3
        MOV  A,R2
        POP  03H
        POP  02H
        RET
;#####################################
; R2,R1,R0 + R7,A = R2,R1,R0
ADD24_16C: PUSH  03H
        PUSH  04H
        MOV   03H,A
        MOV   04H,R7
        CLR   C
        ADDC  A,R0
        MOV   R0,A
        MOV   A,R7
        ADDC  A,R1
        MOV   R1,A
        CLR   A
        ADDC  A,R2
        MOV   R2,A
        MOV   A,R3
        MOV   07H,R4
        POP   04H
        POP   03H
        RET
;#####################################
;(R2,R1,R0 + R7,R6,R5) = R2,R1,R0
ADD24_24C: PUSH  03H
        PUSH  04H
        PUSH  B
        MOV   03H,R5  ;KEEP
        MOV   04H,R6  ;KEEP
        MOV   B,R7    ;KEEP
        CLR   C
        MOV   A,R5    ; R0+R5+C = R0
        ADDC  A,R0
        MOV   R0,A
        MOV   A,R6
        ADDC  A,R1    ; R1+R6+C = R1
        MOV   R1,A
        MOV   A,R7
        ADDC  A,R2
        MOV   R2,A    ; R2+R7+C = R2
        MOV   05H,R3  ; KEEP
        MOV   06H,R4  ; KEEP
        MOV   R7,B    ; KEEP
        POP   B
        POP   04H
        POP   03H
        RET
;#####################################
;INPUT  R0*1,R1*10 - A
;OUTPUT R0,R1
SUB16_8:   CLR  C
        XCH  A,R0
        SUBB A,R0
        JNC  LASTS
        CJNE R1,#00H,DEC
        SETB C
        SJMP LASTS
DEC:    DEC  R1
        CLR  C
LASTS:  XCH  A,R0
        RET
;#####################################
;R1,R0 - R6 = R1,R0
;R6:NOT DISTROYS
SUB16_8C:  CLR  C
        MOV  A,R0
        SUBB A,R6
        MOV  R0,A
        MOV  A,R1
        SUBB A,#00H
        MOV  R1,A
        RET
;#####################################
;R1,R0 - R7,R6 = R1,R0
;R7,R6:NOT DISTROYS
SUB16_16C: CLR  C
        MOV  A,R0
        SUBB A,R6
        MOV  R0,A
        MOV  A,R1
        SUBB A,R7
        MOV  R1,A
        RET
;#####################################
;INPUT  (R0*1,R1*10) * A
;OUTPUT R0,R1
MUL16_8:   PUSH  B
        PUSH  02H
        PUSH  03H
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
          PUSH  07H
          JNZ   NEXT          ;A>0
          MOV   R0,#00H
          MOV   R1,#00H
          SJMP  LASTM
NEXT:     DEC   A
          CJNE  A,#00H,ElseM ;A=A-1
          SJMP  LASTM
ElseM:    MOV   B,A           ;B=COUNTER
          MOV   02H,R0
          MOV   03H,R1
LOOPM:    MOV   A,R2
          MOV   07H,R3
          ACALL ADD16_16
          DJNZ  B,LOOPM
LASTM:    POP   07H
          POP   03H
          POP   02H
          POP   B
          RET
;###############################################
;A     = MULTIPLIER
;R6    = MULTIPLICAND  ;R7=RESEVER
;R1,R0 = PRODUCT
 MULT8C:  PUSH 02H
          PUSH 03H
          PUSH 04H
          PUSH 07H
          MOV  R3,A
          MOV  04H,R6
          MOV  R2,A
          MOV  R0,#00H
          MOV  R1,#00H
          MOV  R7,#00H
          CLR  MULCY
MUL8C2:   MOV   A,R2
          CLR   C
          RRC   A
          MOV   R2,A
          JNC   MUL8C
          MOV   A,R6       ;R1,R0+R7,A=R1,R0
          ACALL ADD16_16C ;RESULT IN R1,R0
          JNC   MUL8C
          SETB  MULCY
MUL8C:    MOV  A,R6        ;SHIFT LEFT OF R7
          CLR  C
          RLC  A
          MOV  R6,A
          MOV  A,R7        ;SHIFT LEFT OF R6
          RLC  A
          MOV  R7,A
          CJNE R2,#00H,MUL8C2
          MOV  C,MULCY
          MOV  06H,R4
          MOV  A,R3
          POP  07H
          POP  04H
          POP  03H
          POP  02H
          RET
```

```
          PUSH  07H
          JNZ   NEXT          ;A>0
          MOV   R0,#00H
          MOV   R1,#00H
          SJMP  LASTM
NEXT:     DEC   A
          CJNE  A,#00H,ElseM ;A=A-1
          SJMP  LASTM
ElseM:    MOV   B,A           ;B=COUNTER
          MOV   02H,R0
          MOV   03H,R1
LOOPM:    MOV   A,R2
          MOV   07H,R3
          ACALL ADD16_16
          DJNZ  B,LOOPM
LASTM:    POP   07H
          POP   03H
          POP   02H
          POP   B
          RET
;###############################################
;A     = MULTIPLIER
;R6    = MULTIPLICAND  ;R7=RESEVER
;R1,R0 = PRODUCT
 MULT8C:  PUSH 02H
          PUSH 03H
          PUSH 04H
          PUSH 07H
          MOV  R3,A
          MOV  04H,R6
          MOV  R2,A
          MOV  R0,#00H
          MOV  R1,#00H
          MOV  R7,#00H
          CLR  MULCY
MUL8C2:   MOV   A,R2
          CLR   C
          RRC   A
          MOV   R2,A
          JNC   MUL8C
          MOV   A,R6       ;R1,R0+R7,A=R1,R0
          ACALL ADD16_16C ;RESULT IN R1,R0
          JNC   MUL8C
          SETB  MULCY
MUL8C:    MOV  A,R6        ;SHIFT LEFT OF R7
          CLR  C
          RLC  A
          MOV  R6,A
          MOV  A,R7        ;SHIFT LEFT OF R6
          RLC  A
          MOV  R7,A
          CJNE R2,#00H,MUL8C2
          MOV  C,MULCY
          MOV  06H,R4
          MOV  A,R3
          POP  07H
          POP  04H
          POP  03H
          POP  02H
          RET
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
;#####################################
;A        =  MULTIPLIER
;R6,R5    =  MULTIPLIER  R7-RESERVE
;R2,R1,R0 =  PRODUCT
MULT16C:   PUSH  03H  ;RESERVE MUL
           PUSH  04H
           PUSH  07H
           PUSH  B
           MOV   04H,R5     ;KEEP
           MOV   B,R6       ;KEEP
           MOV   R0,#00H
           MOV   R1,#00H
           MOV   R2,#00H
           MOV   R7,#00H
           CLR   MULCY
           MOV   R3,A
MUL16C2:   CLR   C
           MOV   A,R3
           RRC   A
           MOV   R3,A
           JNC   MUL16C
           ACALL ADD24_24C
           JNC   MUL16C
           SETB  MULCY
           MOV   R2,#0FFH
           MOV   R1,#0FFH
           MOV   R0,#0FFH
           SJMP  endMul
MUL16C:    CLR  C
           MOV  A,R5 ;SHIFT LEFT OF R5
           RLC  A
           MOV  R5,A
           MOV  A,R6 ;SHIFT LEFT OF R6
           RLC  A
           MOV  R6,A
           MOV  A,R7 ;SHIFT LEFT OF R7
           RLC  A
           MOV  R7,A
           CJNE R3,#00H,MUL16C2
           MOV  05H,R4              ;GET
           MOV  06H,B               ;GET
ENDMUL:    MOV  C,MULCY
           POP  B
           POP  07H
           POP  04H
           POP  03H
           RET
;#############################################
;A          = MULTIPLIER
;R6,R5      = MULTIPLICAND  R7-RESERVE
;R3,R2,R1,R0= PRODUCT
MULT16C2:  PUSH 04H
           PUSH 07H
           PUSH B
           MOV 04H,R5 ;KEEP
           MOV B,R6   ;KEEP
           MOV R0,#00H
           MOV R1,#00H
           MOV R2,#00H
           MOV R7,#00H
           SETB PSW_3
           MOV R0,#00H      ;USE R0 IN BANK1
           CLR PSW_3
           CLR MULCY
           MOV R3,A
MUL16C3:   CLR C
           MOV A,R3
           RRC A
           MOV R3,A
           JNC MUL16_3C
           ACALL ADD24_24C ;R2,R1,R0+R7,R6,R5=R1,R
           SETB PSW_3
           MOV A,R1         ;TO BANK1
           ADDC A,R0
           MOV R1,A
           CLR PSW_3        ;TO BANK0
           JNC MUL16_3C
           SETB MULCY
           MOV R3,#0FFH
           MOV R2,#0FFH
           MOV R1,#0FFH
           MOV R0,#0FFH
           SJMP ENDMUL1
MUL16_3C:  CLR C
           MOV A,R5          ;SHIFT LEFT OF R5
           RLC A
           MOV R5,A
           MOV A,R6          ;SHIFT LEFT OF R6
           RLC A
           MOV R6,A
           MOV A,R7          ;SHIFT LEFT OF R7
           RLC A
           MOV R7,A
           CJNE R3,#00H,MUL16C3
           MOV 05H,R4  ;GET
           MOV R6,B    ;GET
           SETB PSW_3  ;TO BANK1
           MOV A,R1
           CLR PSW_3   ;TO BANK0
           MOV R3,A
ENDMUL1:   MOV C,MULCY
           POP B
           POP 07H
           POP 04H
           RET
;#####################################
;A,R4    = MULTIPLIER   R4*10,A*1
;R6,R5   = MULTIPLICAND  R7=RESEVE   DISTROYS
;R2,R1,R0= PRODUCT
MULT16_16C:PUSH 03H
           PUSH 07H
           MOV R0,#00H
           MOV R1,#00H
           MOV R2,#00H
           MOV R7,#00H
           CLR MULCY    ;CLEAR MULTIPLY FLAG
           MOV R3,A
MUL2C2:    CLR C
           MOV A,R4
           RRC A
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        MOV R2,A
        MOV A,R3
        RLC A
        MOV R3,A        ;SHIFT 0 INTO LSB
        DJNZ B,REVIEW
        MOV R0,02H
        MOV R1,03H
        POP 06H
        POP 03H
        POP 02H
        RET
;################################################
;DIVISOR =R7
;DIVIDEND=R2,R1.R0,QUTTIENT R2,R1,R0
DIV24:  PUSH 03H
        PUSH 04H
        PUSH 05H
        PUSH B
        MOV R5,02H
        MOV R2,#00H
        MOV R3,#00H
        MOV R4,#00H     ;0->QUOTN
        MOV B,#24D      ;24 BIT
        SETB PSW_3      ;TO BANK1
        MOV R0,#00H     ;BUFF OF ACC
        MOV R1,#00H
        CLR PSW_3       ;TO BANK0
REVIEW1: CLR C
        MOV A,R0
        RLC A
        MOV R0,A
        MOV A,R1
        RLC A
        MOV R1,A        ;SHIFT MSB
        MOV A,R5
        RLC A
        MOV R5,A
        SETB PSW_3      ;TO BANK 1
        MOV A,R0        ;ACC
        RLC A
        MOV R0,A
        MOV A,R1        ;ACC
        RLC A
        MOV R1,A
        MOV A,R0        ;KEEP FOR MOMENT
        MOV R2,A
        MOV A,R1
        MOV R3,A
        CLR PSW_3       ;TO BANK 0
        MOV A,R7
        SETB PSW_3      ;TO BANK 1
        MOV R7,A        ;R7 BANK 1=R7 BANK 0
        CLR C
        MOV A,R0
        SUBB A,R7       ;ACC<DIVISER IF YES
        MOV R0,A
        MOV A,R1
        SUBB A,#00H
        MOV R1,A
        JC SHIFT0_1
        SETB C
        SJMP NOTCRY1
SHIFT0_1: CLR C
        MOV A,R3        ;RETURN VALU
        MOV R1,A
        MOV A,R2
        MOV R0,A
NOTCRY1: CLR PSW_3      ;TO BANK 0
        MOV A,R2
        RLC A
        MOV R2,A
        MOV A,R3
        RLC A
        MOV R3,A
        MOV A,R4
        RLC A
        MOV R4,A        ;SHIFT 0
        DJNZ B,REVIEW1  ;COUNT-1=COU
        MOV R0,02H
        MOV R1,03H
        MOV R2,04H
        POP B
        POP 05H
        POP 04H
        POP 03H
        RET
;###############################
;INPUT  R2:R1:R0
;OUTPUT R0
OVERLIMIT1: CLR C
        CLR A
        SUBB A,R2
        JNC  CHKIO
        MOV  R0,#0FFH
        SJMP OUTCHK1
CHKIO:  CLR  C
        CLR  A
        SUBB A,R1
        JNC  OUTCHK1
        MOV  R0,#0FFH
OUTCHK1: NOP
        MOV  A,#NO_DIGIT
;###############################
;INPUT  R2:R1:R0
;OUTPUT R1:R0
OVERLIMIT2:  CLR  C
        CLR  A
        SUBB A,R2
        JNC  OUTCHKPO
        MOV  R0,#0FFH
        MOV  R1,#0FFH
        MOV  R2,#00H
OUTCHKPO:  RET
;#######################################
;INPUT  R3:R2:R1:R0
;OUTPUT R1:R0
OVERLIMIT2_1: CLR  C
        CLR  A
        SUBB A,R2
        JC   MAXX1
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        MOV R4,A  ;R4->A->C
        MOV A,R3
        RRC  A
        MOV R3,A
        JNC MUL2C
        ACALL ADD24_24C
        JNC MUL2C
        SETB MULCY
        MOV R0,#0FFH
        MOV R1,#0FFH
        MOV R2,#0FFH
        SJMP ENDMUL2
MUL2C:  CLR C
        MOV A,R5 ;SHIFT LEFT OF R5
        RLC A
        MOV R5,A
        MOV A,R6 ;SHIFT LEFT OF R6
        RLC A
        MOV R6,A
        MOV A,R7 ;SHIFT LEFT OF R7
        RLC A
        MOV R7,A
        CJNE R4,#00H,MUL2C2
        CJNE R3,#00H,MUL2C2
ENDMUL2: MOV C,MULCY
        POP 07H
        POP 03H
        RET
;#################################
;A,R4=MULTIPLIER R4*10,A*1
;R6,R5=MULTIPLICAND R7-RESERVE DISTROYS
;R3,R2,R1,R0=PRODUCT
MULT16_16C2:PUSH 07H
        MOV R0,#00H
        MOV R1,#00H
        MOV R2,#00H
        MOV R7,#00H
        SETB PSW_3
        MOV R0,#00H  ;USE R0 IN BANK 1
        MOV R1,#00H  ;USE R1 IN BANK 1
        CLR PSW_3
        CLR MULCY    ;CLEAR MULTIPLY FLAG
        MOV R3,A
MUL3C2: CLR C
        MOV A,R4
        RRC A
        MOV R4,A     ;R4->A->C
        MOV A,R3
        RRC A
        MOV R3,A
        JNC MUL3C
        ACALL ADD24_24C ;R2,R1,R0+(R0)
        JNC MUL3C
        SETB PSW_3
        MOV A,R1     ;TO BANK 1
        ADDC A,R0
        MOV R1,A
        CLR PSW_3    ;TO BANK 0
        JNC MUL3C
        SETB MULCY  ;SET MULTIPLY BIT
        MOV R0,#0FFH
        MOV R1,#0FFH
        MOV R2,#0FFH
        MOV R3,#0FFH
        SJMP ENDMUL3
MUL3C:  CLR C
        MOV A,R5    ;SHIFT LEFT OF R5
        RLC A
        MOV R5,A
        MOV A,R6    ;SHIFT LEFT OF R6
        RLC A
        MOV R6,A
        MOV A,R7    ;SHIFT LEFT OF R7
        RLC A
        MOV R7,A
        SETB PSW_3  ;BANK 1
        MOV A,R0
        RLC A
        MOV R0,A
        CLR PSW_3   ;BANK 0
        CJNE R4,#00H,MUL3C2
        CJNE R3,#00H,MUL3C2
        SETB PSW_3   ;TO BANK 1
        MOV A,R1
        CLR PSW_3    ;TO BANK 0
        MOV R3,A     ;R0 BK1->R3
ENDMUL3: MOV C,MULCY
        POP 07H
        RET
;#################################################
;INPUT   = R1:R0
;DIVISOR = R7
;DIVIDEND= R1,R0,QUTTIENT
DIV16:  PUSH 02H
        PUSH 03H
        PUSH 06H
        MOV R2,#00H
        MOV R3,#00H     ;0->QUOTN
        MOV B,#16D      ;16 BIT
        CLR A
        MOV R6,A        ;BUFF OF ACC
REVIEW: CLR C
        MOV A,R0
        RLC A
        MOV R0,A
        MOV A,R1
        RLC A
        MOV R1,A        ;SHIFT MSB OF DIVND
        MOV A,R6
        RLC A
        MOV R6,A
        CLR C
        SUBB A,R7
        JC SHIFT0
        MOV R6,A
        SETB C
        SJMP NOTCRY
SHIFT0: CLR C
NOTCRY: MOV A,R2
        RLC A
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            CLR  A
            SUBB A,R3
            JNC  OUTCHKPO_1
MAXX1:      MOV  R0,#0FFH
            MOV  R1,#0FFH
            MOV  R2,#00H
OUTCHKPO_1: RET
;###############################
;INPUT  R2:R1:R0
;OUTPUT R1:R0
OVERLIMIT_KV:
            CLR  C
            MOV  A,#1AH
            SUBB A,R0
            JNC  OUTCHKPO_KV
            MOV  R0,#1AH
               ;MOV  R1,#0FFH
               ;MOV  R2,#00H
OUTCHKPO_KV:  NOP
            RET
;###############################
;INPUT  R2:R1:R0
;SHIFT LEFT R7 TIME
SHIFTDIG:   MOV  A,R7
            JZ   ENDSHIF
SHIFTLOOP:  MOV  A,R2
            CLR  C
            RRC  A
            MOV  R2,A
            MOV  A,R1
            RRC  A
            MOV  R1,A
            MOV  A,R0
            RRC  A
            MOV  R0,A
            DJNZ R7,SHIFTLOOP
ENDSHIF:    RET
;################################
;INPUT  R2:R1:R0
;SHIFT LEFT R7 TIME
SHIFTDIG2:  MOV  A,R7
            JZ   ENDSHIF2
SHIFTLOOP2: CLR  C
            MOV  A,R3
            RRC  A
            MOV  R3,A
            MOV  A,R2
            RRC  A
            MOV  R2,A
            MOV  A,R1
            RRC  A
            MOV  R1,A
            MOV  A,R0
            RRC  A
            MOV  R0,A
            DJNZ R7,SHIFTLOOP2
ENDSHIF2:   RET
;################################
DELAY1S:  PUSH   02H
            MOV  R2,#0FFH       ; TIME 0.25 SEC
            LCALL DELAY2
            MOV  R2,#0FFH       ; TIME 0.25 SEC
            LCALL DELAY2
            MOV  R2,#0FFH       ; TIME 0.25 SEC
            LCALL DELAY2
            MOV  R2,#0FFH       ; TIME 0.25 SEC
            LCALL DELAY2
            POP  02H
            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,$+3
            JNC   HTOAS1
            ORL   A,#30H
            RET
HTOAS1:     SUBB  A,#9
            ORL   A,#40H
            RET
; ********** SBYTE SUB **********
; SEND BYTE
; IN  = A
; REG = NO
SBYTE:      JNB   TI,$              ;WAIT FOR SEND OK
            CLR   TI
            MOV   SBUF,A
            RET
; .********** SBYTEH SUB **********
; SEND 2 BYTE HEX FROM A
; IN  = A
; REG = A,R2,R3

SBYTEH:     LCALL HTOA
            MOV   A,R2
            LCALL SBYTE
            MOV   A,R3
            LCALL SBYTE
            RET
; ********** SBLANK SUB **********
; SEND BLANK
; IN  = R2 (NO OF BLANK)
; REG = A,R2

SBLANK:     ;PUSH   ACC
            MOV   A,#' '
            LCALL SBYTE
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        DJNZ    R2,SBLANK
        ;POP     ACC
        RET
; ********** SLF SUB **********
; PRINT CR/LF
; REG = A
SLF:    ;PUSH    ACC
        MOV     A,#0DH
        LCALL   SBYTE
        MOV     A,#0AH
        LCALL   SBYTE
        ;POP     ACC
        RET
; ********** Sxxx SUB **********
S1B:    ;PUSH    ACC
        MOV     A,#' '
        LCALL   SBYTE
        ;POP     ACC
        RET
; ********** DTSEC SUB **********
; DELAY 1/10 SECOND
; IN  = R2
; REG = R2,R3,R4
DTSEC:  MOV     R3,#179
DTSEC1: MOV     R4,#0
        DJNZ    R4,$
        NOP
        NOP
        DJNZ    R3,DTSEC1
        DJNZ    R2,DTSEC
        RET
;=====================================
;DELAY DURATION
DELAY2:     PUSH B
DE_LOOP1:   MOV  B,#0FFH
DE_LOOP2:   DJNZ B,DE_LOOP2
            DJNZ R2,DE_LOOP1
            POP  B
            RET
;-------------------------------
WriteAD:    PUSH    00H
            PUSH    01H
            PUSH    02H
            PUSH    03H
            PUSH    04H
            PUSH    05H
            PUSH    06H
            PUSH    07H
            PUSH    ACC
            CLR     C
            CLR     ACC
            MOV     R0,#00H
            MOV     R1,#00H
            MOV     R7,#0AH
Loopbf:     MOV     DPTR,#IC_ADC
            MOV     A,#00H
            MOVX    @DPTR,A
            MOV     R2,#01H
            LCALL   DELAY2
            MOV     DPTR,#IC_ADC
            MOVX    A,@DPTR         ; read input
            MOV     R6,A            ; save input
            LCALL   ADD16_8C        ;(R1R0 + R6)
            DJNZ    R7,Loopbf
            MOV     R7,#0AH
            LCALL   DIV16           ;R1R0/R7->R1:R0
            MOV     ADBuff,R0
OUTAD:      NOP
            POP     ACC
            POP     07H
            POP     06H
            POP     05H
            POP     04H
            POP     03H
            POP     02H
            POP     01H
            POP     00H
            RET
;*********************************
;      SUB.  DISPLAY LED FLAG     *
;*********************************
DSP_FLAG:   MOV     DPTR,#P_DSP     ; set flag active
            MOV     A,#FLAG
            MOVX    @DPTR,A         ; enable flag
            MOV     DPTR,#D_DSP
            MOV     A,ADBUFF            ; get input
            MOVX    @DPTR,A         ; out data to flag
            RET
;*************************************
;         DAC0832                    *
;*************************************
OUTPUT:     MOV     DPTR,#IC_DAC
            MOV     A,DABuff
            MOVX    @DPTR,A
            RET
            ;*********************************
;=================================================
;AUTO START
;=================================================
        ;ORG  0DFFFH
        ;DB   0A3H
        END
;#################################################
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
########################################
; include for data LCD_PROJECT SIMULATOR

          ORG     8000H
          JMP     START
          ORG     8008H
PER00:     DB   '=> 0.0 %'
PER01:     DB   '=> 0.0 %'
PER02:     DB   '=> 1.0 %'
PER03:     DB   '=> 1.0 %'
PER04:     DB   '=> 1.5 %'
PER05:     DB   '=> 2.0 %'
PER06:     DB   '=> 2.5 %'
PER07:     DB   '=> 2.5 %'
PER08:     DB   '=> 3.0 %'
PER09:     DB   '=> 4.0 %'
PER0A:     DB   '=> 4.5 %'
PER0B:     DB   '=> 4.5 %'
PER0C:     DB   '=> 5.1 %'
PER0D:     DB   '=> 5.5 %'
PER0E:     DB   '=> 6.0 %'
PER0F:     DB   '=> 6.0 %'
PER10:     DB   '=> 6.5 %'
PER11:     DB   '=> 7.0 %'
PER12:     DB   '=> 7.5 %'
PER13:     DB   '=> 8.0 %'
PER14:     DB   '=> 8.0 %'
PER15:     DB   '=> 8.5 %'
PER16:     DB   '=> 9.0 %'
PER17:     DB   '=> 9.5 %'
PER18:     DB   '=>10.0 %'
PER19:     DB   '=>10.0 %'
PER1A:     DB   '=>10.5 %'
PER1B:     DB   '=>11.0 %'
PER1C:     DB   '=>11.5 %'
PER1D:     DB   '=>12.0 %'
PER1E:     DB   '=>12.0 %'
PER1F:     DB   '=>12.5 %'
PER20:     DB   '=>13.0 %'
PER21:     DB   '=>13.5 %'
PER22:     DB   '=>13.5 %'
PER23:     DB   '=>14.0 %'
PER24:     DB   '=>14.5 %'
PER25:     DB   '=>15.0 %'
PER26:     DB   '=>15.5 %'
PER27:     DB   '=>15.5 %'
PER28:     DB   '=>16.0 %'
PER29:     DB   '=>16.5 %'
PER2A:     DB   '=>17.0 %'
PER2B:     DB   '=>17.0 %'
PER2C:     DB   '=>17.5 %'
PER2D:     DB   '=>18.0 %'
PER2E:     DB   '=>18.5 %'
PER2F:     DB   '=>19.0 %'
PER30:     DB   '=>19.0 %'
PER31:     DB   '=>19.5 %'
PER32:     DB   '=>20.0 %'
PER33:     DB   '=>20.5 %'
PER34:     DB   '=>21.0 %'
PER35:     DB   '=>21.0 %'
PER36:     DB   '=>21.5 %'
PER37:     DB   '=>22.0 %'
PER38:     DB   '=>22.5 %'
PER39:     DB   '=>22.5 %'
PER3A:     DB   '=>23.0 %'
PER3B:     DB   '=>23.5 %'
PER3C:     DB   '=>24.0 %'
PER3D:     DB   '=>24.5 %'
PER3E:     DB   '=>24.5 %'
PER3F:     DB   '=>25.0 %'
PER40:     DB   '=>25.5 %'
PER41:     DB   '=>26.0 %'
PER42:     DB   '=>26.5 %'
PER43:     DB   '=>26.5 %'
PER44:     DB   '=>27.0 %'
PER45:     DB   '=>27.5 %'
PER46:     DB   '=>28.0 %'
PER47:     DB   '=>28.0 %'
PER48:     DB   '=>28.5 %'
PER49:     DB   '=>29.0 %'
PER4A:     DB   '=>29.5 %'
PER4B:     DB   '=>30.0 %'
PER4C:     DB   '=>30.0 %'
PER4D:     DB   '=>30.5 %'
PER4E:     DB   '=>31.0 %'
PER4F:     DB   '=>31.5 %'
PER50:     DB   '=>32.0 %'
PER51:     DB   '=>32.0 %'
PER52:     DB   '=>32.5 %'
PER53:     DB   '=>33.0 %'
PER54:     DB   '=>33.5 %'
PER55:     DB   '=>33.5 %'
PER56:     DB   '=>34.0 %'
PER57:     DB   '=>34.5 %'
PER58:     DB   '=>35.0 %'
PER59:     DB   '=>35.5 %'
PER5A:     DB   '=>35.5 %'
PER5B:     DB   '=>36.0 %'
PER5C:     DB   '=>36.5 %'
PER5D:     DB   '=>37.0 %'
PER5E:     DB   '=>37.0 %'
PER5F:     DB   '=>37.5 %'
PER60:     DB   '=>38.0 %'
PER61:     DB   '=>38.5 %'
PER62:     DB   '=>38.5 %'
PER63:     DB   '=>39.0 %'
PER64:     DB   '=>39.0 %'
PER65:     DB   '=>39.5 %'
PER66:     DB   '=>40.0 %'
PER67:     DB   '=>40.5 %'
PER68:     DB   '=>41.0 %'
PER69:     DB   '=>41.0 %'
PER6A:     DB   '=>41.5 %'
PER6B:     DB   '=>42.0 %'
PER6C:     DB   '=>42.5 %'
PER6D:     DB   '=>42.5 %'
PER6E:     DB   '=>43.0 %'
PER6F:     DB   '=>43.5 %'
PER70:     DB   '=>44.0 %'
PER71:     DB   '=>44.5 %'
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
PER72:      DB  '=>44.5 %'
PER73:      DB  '=>45.0 %'
PER74:      DB  '=>45.5 %'
PER75:      DB  '=>46.0 %'
PER76:      DB  '=>46.0 %'
PER77:      DB  '=>46.5 %'
PER78:      DB  '=>47.0 %'
PER79:      DB  '=>47.5 %'
PER7A:      DB  '=>48.0 %'
PER7B:      DB  '=>48.0 %'
PER7C:      DB  '=>48.5 %'
PER7D:      DB  '=>49.0 %'
PER7E:      DB  '=>49.5 %'
PER7F:      DB  '=>50.0 %'
PER80:      DB  '=>50.0 %'
PER81:      DB  '=>50.5 %'
PER82:      DB  '=>51.0 %'
PER83:      DB  '=>51.5 %'
PER84:      DB  '=>52.0 %'
PER85:      DB  '=>52.5 %'
PER86:      DB  '=>52.5 %'
PER87:      DB  '=>53.0 %'
PER88:      DB  '=>53.5 %'
PER89:      DB  '=>53.5 %'
PER8A:      DB  '=>54.0 %'
PER8B:      DB  '=>54.5 %'
PER8C:      DB  '=>55.0 %'
PER8D:      DB  '=>55.0 %'
PER8E:      DB  '=>55.5 %'
PER8F:      DB  '=>55.5 %'
PER90:      DB  '=>56.5 %'
PER91:      DB  '=>57.0 %'
PER92:      DB  '=>57.0 %'
PER93:      DB  '=>57.5 %'
PER94:      DB  '=>58.0 %'
PER95:      DB  '=>58.5 %'
PER96:      DB  '=>59.0 %'
PER97:      DB  '=>59.0 %'
PER98:      DB  '=>60.0 %'
PER99:      DB  '=>60.0 %'
PER9A:      DB  '=>60.5 %'
PER9B:      DB  '=>61.0 %'
PER9C:      DB  '=>61.0 %'
PER9D:      DB  '=>61.5 %'
PER9E:      DB  '=>62.0 %'
PER9F:      DB  '=>62.5 %'
PERA0:      DB  '=>62.5 %'
PERA1:      DB  '=>63.0 %'
PERA2:      DB  '=>63.5 %'
PERA3:      DB  '=>64.0 %'
PERA4:      DB  '=>64.5 %'
PERA5:      DB  '=>64.5 %'
PERA6:      DB  '=>65.0 %'
PERA7:      DB  '=>65.5 %'
PERA8:      DB  '=>66.0 %'
PERA9:      DB  '=>66.5 %'
PERAA:      DB  '=>66.5 %'
PERAB:      DB  '=>67.0 %'
PERAC:      DB  '=>67.5 %'
PERAD:      DB  '=>68.0 %'
PERAE:      DB  '=>68.0 %'
PERAF:      DB  '=>68.5 %'
PERB0:      DB  '=>69.0 %'
PERB1:      DB  '=>69.5 %'
PERB2:      DB  '=>70.0 %'
PERB3:      DB  '=>70.0 %'
PERB4:      DB  '=>70.5 %'
PERB5:      DB  '=>71.0 %'
PERB6:      DB  '=>71.5 %'
PERB7:      DB  '=>72.0 %'
PERB8:      DB  '=>72.0 %'
PERB9:      DB  '=>72.5 %'
PERBA:      DB  '=>73.0 %'
PERBB:      DB  '=>73.5 %'
PERBC:      DB  '=>74.0 %'
PERBD:      DB  '=>74.0 %'
PERBE:      DB  '=>74.5 %'
PERBF:      DB  '=>75.0 %'
PERC0:      DB  '=>75.5 %'
PERC1:      DB  '=>75.5 %'
PERC2:      DB  '=>76.0 %'
PERC3:      DB  '=>76.5 %'
PERC4:      DB  '=>77.0 %'
PERC5:      DB  '=>77.0 %'
PERC6:      DB  '=>77.5 %'
PERC7:      DB  '=>78.0 %'
PERC8:      DB  '=>78.5 %'
PERC9:      DB  '=>79.0 %'
PERCA:      DB  '=>79.0 %'
PERCB:      DB  '=>79.0 %'
PERCC:      DB  '=>80.0 %'
PERCD:      DB  '=>80.5 %'
PERCE:      DB  '=>81.0 %'
PERCF:      DB  '=>81.0 %'
PERD0:      DB  '=>81.5 %'
PERD1:      DB  '=>82.0 %'
PERD2:      DB  '=>82.5 %'
PERD3:      DB  '=>82.5 %'
PERD4:      DB  '=>83.0 %'
PERD5:      DB  '=>83.5 %'
PERD6:      DB  '=>84.0 %'
PERD7:      DB  '=>84.5 %'
PERD8:      DB  '=>85.0 %'
PERD9:      DB  '=>85.0 %'
PERDA:      DB  '=>85.5 %'
PERDB:      DB  '=>86.0 %'
PERDC:      DB  '=>86.5 %'
PERDD:      DB  '=>86.5 %'
PERDE:      DB  '=>87.0 %'
PERDF:      DB  '=>87.5 %'
PERE0:      DB  '=>88.0 %'
PERE1:      DB  '=>88.0 %'
PERE2:      DB  '=>88.5 %'
PERE3:      DB  '=>89.0 %'
PERE4:      DB  '=>89.5 %'
PERE5:      DB  '=>90.0 %'
PERE6:      DB  '=>90.0 %'
PERE7:      DB  '=>90.5 %'
PERE8:      DB  '=>90.5 %'
PERE9:      DB  '=>91.0 %'
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
 PEREA:      DB  '=>91.5 %'
PEREB:       DB  '=>92.0 %'
PEREC:       DB  '=>92.5 %'
PERED:       DB  '=>93.0 %'
PEREE:       DB  '=>93.5 %'
PEREF:       DB  '=>94.0 %'
PERF0:       DB  '=>94.0 %'
PERF1:       DB  '=>94.5 %'
PERF2:       DB  '=>95.0 %'
PERF3:       DB  '=>95.5 %'
PERF4:       DB  '=>95.5 %'
PERF5:       DB  '=>96.0 %'
PERF6:       DB  '=>96.5 %'
PERF7:       DB  '=>97.0 %'
PERF8:       DB  '=>97.0 %'
PERF9:       DB  '=>97.5 %'
PERFA:       DB  '=>98.0 %'
PERFB:       DB  '=>98.5 %'
PERFC:       DB  '=>99.0 %'
PERFD:       DB  '=>99.0 %'
PERFE:       DB  '=>99.5 %'
PERFF:       DB  '=>99.9 %'
PERFFF:      DB  '=>99.9 %'
FORNTSW:    DB  'Const =>'
TITLE1:     DB  'PROCESS-'
TITLE2:     DB  'SIMULATE'
K1_gain:    DB  ' K1_gain'
K2_gain:    DB  ' K2_gain'
K3_gain:    DB  ' K3_gain'
T1_Time:    DB  ' T1_time'
T2_Time:    DB  ' T2_time'
T3_Time:    DB  ' T3_time'
SP_Per:     DB  'MV_Valve'
TI_TEMP:    DB  ' TEMP    '
FS_FLOW:    DB  ' FLOW    '
ThermalSW1: DB  'MODE The'
ThermalSW2: DB  'rmal:Y/N'
LevelSW1:   DB  'MODE Lev'
LevelSW2:   DB  'el  :Y/N'
GasSW1:     DB  'MODE Gas'
GasSW2:     DB  '    :Y/N'
ChemicalSW1:DB  'MODE Che'
ChemicalSW2:DB  'mic :Y/N'
Thermal_TITLE1:  DB  'Thermal '
Level_TITLE1:    DB  ' Level  '
Gas_TITLE1:      DB  '  Gas   '
Chemical_TITLE1: DB  'Chemical'
Simul_TITLE:     DB  'Simulate'
;=====================================================================
        TableDSP2:
;=====================================================================
;       ADDRESS FOR LCD Index_LSB --> Show Percent %
;=====================================================================
DB 008H,008H,010H,018H,020H,028H,030H,038H,040H,048H,050H,058H,060H,068H,070H,078H
DB 080H,088H,090H,098H,0A0H,0A8H,0B0H,0B8H,0C0H,0C8H,0D0H,0D8H,0E0H,0E8H,0F0H,0F8H
DB 000H,008H,010H,018H,020H,028H,030H,038H,040H,048H,050H,058H,060H,068H,070H,078H
DB 080H,088H,090H,098H,0A0H,0A8H,0B0H,0B8H,0C0H,0C8H,0D0H,0D8H,0E0H,0E8H,0F0H,0F8H
DB 000H,008H,010H,018H,020H,028H,030H,038H,040H,048H,050H,058H,060H,068H,070H,078H
DB 080H,088H,090H,098H,0A0H,0A8H,0B0H,0B8H,0C0H,0C8H,0D0H,0D8H,0E0H,0E8H,0F0H,0F8H
DB 000H,008H,010H,018H,020H,028H,030H,038H,040H,048H,050H,058H,060H,068H,070H,078H
```

เอกสารแนบเอกสารทางวิชาการเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ขอขอบคุณ ผู้ที่ได้ทำการเผยแพร่ และขอขอบคุณ ผู้ที่ได้ทำการเผยแพร่เอกสารที่มีการนำไปใช้

```
 DB 080H,088H,090H,098H,0A0H,0A8H,0B0H,0B8H,0C0H,0C8H,0D0H,0D8H,0E0H,0E8H,0F0H,0F8H
DB 000H,008H,010H,018H,020H,028H,030H,038H,040H,048H,050H,058H,060H,068H,070H,078H
DB 080H,088H,090H,098H,0A0H,0A8H,0B0H,0B8H,0C0H,0C8H,0D0H,0D8H,0E0H,0E8H,0F0H,0F8H
DB 000H,008H,010H,018H,020H,028H,030H,038H,040H,048H,050H,058H,060H,068H,070H,078H
DB 080H,088H,090H,098H,0A0H,0A8H,0B0H,0B8H,0C0H,0C8H,0D0H,0D8H,0E0H,0E8H,0F0H,0F8H
DB 000H,008H,010H,018H,020H,028H,030H,038H,040H,048H,050H,058H,060H,068H,070H,078H
DB 080H,088H,090H,098H,0A0H,0A8H,0B0H,0B8H,0C0H,0C8H,0D0H,0D8H,0E0H,0E8H,0F0H,0F8H
DB 000H,008H,010H,018H,020H,028H,030H,038H,040H,048H,050H,058H,060H,068H,070H,078H
DB 080H,088H,090H,098H,0A0H,0A8H,0B0H,0B8H,0C0H,0C8H,0D0H,0D8H,0E0H,0E8H,0F0H,0F8H
DB 000H,000H,000H,000H,000H,000H,000H,000H,000H,000H,000H,000H,000H,000H,000H,000H
;=====================================================================
        TableDSP3:
;=====================================================================
;       ADDRESS FOR LCD Index_MSB --> Show Percent %
;=====================================================================
    DB 80H,80H,80H,80H,80H,80H,80H,80H,80H,80H,80H,80H,80H,80H,80H,80H
    DB 80H,80H,80H,80H,80H,80H,80H,80H,80H,80H,80H,80H,80H,80H,80H,80H
    DB 81H,81H,81H,81H,81H,81H,81H,81H,81H,81H,81H,81H,81H,81H,81H,81H
    DB 81H,81H,81H,81H,81H,81H,81H,81H,81H,81H,81H,81H,81H,81H,81H,81H
    DB 82H,82H,82H,82H,82H,82H,82H,82H,82H,82H,82H,82H,82H,82H,82H,82H
    DB 82H,82H,82H,82H,82H,82H,82H,82H,82H,82H,82H,82H,82H,82H,82H,82H
    DB 83H,83H,83H,83H,83H,83H,83H,83H,83H,83H,83H,83H,83H,83H,83H,83H
    DB 83H,83H,83H,83H,83H,83H,83H,83H,83H,83H,83H,83H,83H,83H,83H,83H
    DB 84H,84H,84H,84H,84H,84H,84H,84H,84H,84H,84H,84H,84H,84H,84H,84H
    DB 84H,84H,84H,84H,84H,84H,84H,84H,84H,84H,84H,84H,84H,84H,84H,84H
    DB 85H,85H,85H,85H,85H,85H,85H,85H,85H,85H,85H,85H,85H,85H,85H,85H
    DB 85H,85H,85H,85H,85H,85H,85H,85H,85H,85H,85H,85H,85H,85H,85H,85H
    DB 86H,86H,86H,86H,86H,86H,86H,86H,86H,86H,86H,86H,86H,86H,86H,86H
    DB 86H,86H,86H,86H,86H,86H,86H,86H,86H,86H,86H,86H,86H,86H,86H,86H
    DB 87H,87H,87H,87H,87H,87H,87H,87H,87H,87H,87H,87H,87H,87H,87H,87H
    DB 87H,87H,87H,87H,87H,87H,87H,87H,87H,87H,87H,87H,87H,87H,87H,87H
    DB 88H,88H,88H,88H,88H,88H,88H,88H,88H,88H,88H,88H,88H,88H,88H,88H
;#####################################################################
; GENERAL SUBROUNTINE
;#####################################################################
; SHOWE LCD
ShowLCD:
          LCALL   INITLCD
          MOV     DPTR,#ForntSW
          LCALL   DISP1
          MOV     A,#40H          ;New Address
          LCALL   GOTOXY
          MOV     DPH,R4
          MOV     DPL,R3
          LCALL   DISP0
          RET
;================================================
; SHOWE RIGHT_LCD
DISP1:
           PUSH    00H
           PUSH    ACC
           MOV     R0,#8           ;R0 <==== 8 data
                                   ;MOV     DPTR,#TABLE1
STR1:      MOVX    A,@DPTR         ;Get character
           LCALL   WRITE_CHR       ;Write to LCD
           INC     DPTR            ;Next
           DJNZ    R0,STR1
           POP     ACC
           POP     00H
           RET
```

เอกสารนี้มีลิขสิทธิ์ สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
;=================================================
; SHOWE LEFT LCD
DISP0:
        PUSH    00H
        PUSH    ACC
        MOV     R0,#8           ;R0 <==== 8 data
STT2:   MOVX    A,@DPTR
        LCALL   WRITE_CHR
        INC     DPTR
        DJNZ    R0,STT2
        POP     ACC
        POP     00H
        RET
;=========================================================
INITLCD:  PUSH    DPH
        PUSH    DPL
        PUSH    ACC
        MOV     DPTR,#COMMAND
        MOV     A,#38H          ;8bit, 2 line, 5x7 dot
        MOVX    @DPTR,A         ;Write instruction
        LCALL   WAITBF          ;Wait LCD ready
        MOV     A,#0FH          ;Display-on,cursor-on & blink
        MOVX    @DPTR,A
        LCALL   WAITBF
        MOV     A,#6            ;cursor increment
        MOVX    @DPTR,A
        LCALL   WAITBF
        MOV     A,#1            ;clear display and home
        MOVX    @DPTR,A
        LCALL   WAITBF
        POP     ACC
        POP     DPL
        POP     DPH
        RET
;###############################################
; *** Write ASCII to LCD ***
WRITE_CHR: PUSH    00H
        PUSH    ACC
        PUSH    DPL
        PUSH    DPH
        MOV     DPTR,#WRITEDATA
        MOVX    @DPTR,A
        LCALL   WAITBF      ;Wait LCD module ready
        POP     DPH
        POP     DPL
        POP     ACC
        POP     00H
        RET
; *** Wait for ready ,check busy flag ***
WAITBF:   PUSH    DPL
        PUSH    DPH
        PUSH    ACC
        MOV     DPTR,#READBUSY
RDY1:   MOVX    A,@DPTR
        JB      ACC.7,RDY1      ;Busy Flag
        POP     ACC
        POP     DPH
        POP     DPL
        RET
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
 ;*
;* Go to address of LCD *
;*
GOTOXY:     PUSH    DPH
            PUSH    DPL
            MOV     DPTR,#COMMAND
            SETB    ACC.7           ;Set DDRAM instruction
            MOVX    @DPTR,A
            LCALL   WAITBF          ;Wait busy flag
            POP     DPL
            POP     DPH
            RET
;=========================================
; INDEX FOR TABLE-LCD show Percent %%
;=========================================
        tableLCD:
;=========================================
;INPUT  A :COUNT INDEX
;OUTPUT R0:R1
;       R4:R3 = INDEX LCD --> DPTR
;       R4 = DSP3
;       R3 = DSP2
;=========================================
        PUSH  07H
        MOV   R7,A              ;DATA TO BUFFER
        MOV   DPTR,#TableDSP2
        MOV   A,R7
        MOVC  A,@A+DPTR
        MOV   R3,A
        MOV   DPTR,#TableDSP3
        MOV   A,R7
        MOVC  A,@A+DPTR
        MOV   R4,A
        POP   07H
        RET
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                        /*** disp.c ***/
#include <stdio.h>
#include <conio.h>
#include <stdlib.h>
#include <math.h>
#include <string.h>
#include <dos.h>
#include <stdlib.h>

#include "rtstdhdr.h"
#include "rtgsubs.h"
#include "rtbargr.h"
#include "rtgraph.h"
#include "rtchart.h"

#include "interfac.h"

extern prndriver, prnmode, prnport, orient, formfeed;   /* set printer */
extern cport, baud, parity, stopbits, numbits;          /* set serial  */
extern s_setpoint, s_high_a, s_low_a, d_select;         /* set parameter */

        realtype        ydata[1000], yvalues[6];
        realtype        timeint, sampleint, rt, lalarm, halarm, stpnt;
        realtype        lalarm, halarm, miny1, maxy1, miny2, maxy2,
                        centerp, rnum, halarm1, lalarm1, halarm2, lalarm2,
                        rr, miny, maxy, centerp, rnum;

        tagtype         tags[6];
        rtstatptr       rtstat[5];
        char            ch,units[20],title[40],units1[16], units2[16];      do{
        int             i, j, nt, decs, lc[6], lf[6], decs1, decs2;
        int             grid, ratchf, xdecs, ydecs;

        unsigned int    index;
        alarmstrtype    alarmstrings[3];

        int chn1, chn2, chn3, chn4, chn5, chn6;

void screen();
void testkey(char ch);

void display()
{
      /* INITIALIZE THE GRAPHICS ADAPTER SET UP 5 REAL TIME WINDOWS */
                rtinitgraphics(defaultbgidir, 5, rtstat,1);

switch(d_select)
  {
      case 1:   block();
                break;

      case 2:   screen();
                /* SET WINDOW TO CENTER OF SCREEN */
                rtsetpercentwindow(rtstat[1],0.0,0.15,1.0,0.75);

                /* VALUES FOR LIMITS */
                timeint = 4.0;   sampleint = 0.05;  miny = 0.0;
                maxy = 100.0;       rt = 0.50;        nt = 6;
                grid = 4;         lalarm = s_low_a;  halarm = s_high_a;
                stpnt = s_setpoint;   ratchf =  4;       xdecs = 2;
                ydecs = 1;

                /* SET TAG NAMES */
                strcpy(tags[0],"MV");
                strcpy(tags[1],"PV");
                strcpy(tags[2],"Temp Liquid");
                strcpy(tags[3],"Feed Liquid");
                strcpy(tags[4],"Feed Steam");
                strcpy(tags[5],"Temp Steam");
                strcpy(units,"Percent (%)");
                strcpy(title,"PROCESS SIMULATOR");

                for (i = 0; i < nt; i++){
                    lc[i] = 14-i;
                    lf[i] = 0;
                }
                /* SET UP LINE GRAPH */
                rtsetupscrollgraph(rtstat[1],timeint, sampleint, miny,
                                   maxy, rt,nt, grid, lalarm, halarm,
                                   stpnt, xdecs, ydecs,title, units,
                                   tags, lc, lf, ratchf);

                /* LIMIT WINDOW */
                rtborderwindow(rtstat[1],15);


                /* READ SIGNAL */
                chn1 = r_port(1);
                chn2 = r_port(1);
                chn3 = r_port(1);
                chn4 = r_port(1);
                chn5 = r_port(1);
                chn6 = r_port(1);

                chn1 = chn1 & 0x00ff;
                chn2 = chn2 & 0x00ff;
                chn3 = chn3 & 0x00ff;
                chn4 = chn4 & 0x00ff;
                chn5 = chn5 & 0x00ff;
                chn6 = chn6 & 0x00ff;

                yvalues[0] = (chn1/2.55);
                yvalues[1] = (chn2/2.55);
                yvalues[2] = (chn3/2.55);
                yvalues[3] = (chn4/2.55);
                yvalues[4] = (chn5/2.55);
                yvalues[5] = (chn6/2.55);

                /* UP DATE LINE GRAPH */
                rtupdatescrollgraph(rtstat[1],yvalues);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            }while (kbhit() ==0);

             rtclosegraphics(rtstat,1);

              /* FOR PRINTER (FUTURE) */
                        /*ch = getch();
                           testkey(ch);
                           screen(); */

          break;
    }
}

void screen()
{
               rtsetpercentwindow(rtstat[0],0.0,0.0,1.0,0.14);
                     setcolor(LIGHTMAGENTA);
                     outtextxy(1,10,"PROCESS  SIMULATOR ");
                     outtextxy(1,32,"KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG");
               rtsetpercentwindow(rtstat[3],0.01,0.76,0.79,0.99);
                     setcolor(LIGHTBLUE);
                     outtextxy(40,75,"START AUTO PROGRAM SIMULATION");
                     outtextxy(40,90,"STOP PROGRAM SIMULATION");
                     outtextxy(40,105,"PRINT SIMULATION");
                     outtextxy(20,75,"S");
                     outtextxy(20,90,"E");
                     outtextxy(20,105,"P");

}

void testkey(char ch)
{
    switch(ch)
      {
        case 'P':    rthardcopyreturn(rtstat,0,10,prndriver,prnmode,
                                      prnport,orient,formfeed,"",2,NULL);
                     break;

        case 'p':    rthardcopyreturn(rtstat,0,10,prndriver,prnmode,
                                      prnport,orient,formfeed,"",2,NULL);
                     break;
      }
}

void drawdiagram(void)
{
  rtsetwindow(rtstat[3]);
  rtsetworldcoordinates(0.0, 0.0, 1.0, 1.0);
       /* SET UP BLOCK DIAGRAM */
       rtbartextworld(0.04,0.80, 0.08,0.05,"Kv",    11,0 );
       rtbartextworld(0.13,0.80, 0.08,0.06,"LAG",   11,0 );
       rtbartextworld(0.23,0.80, 0.08,0.06,"Kw",    11,0 );
       rtbartextworld(0.33,0.70, 0.09,0.08,"e(k)",  11,0 );
       rtbartextworld(0.45,0.70, 0.08,0.06,"LAG",   11,0 );
       rtbartextworld(0.55,0.70, 0.08,0.06,"Ks",    11,0 );
       rtbartextworld(0.65,0.60, 0.09,0.08,"e(k)",  11,0 );
       rtbartextworld(0.77,0.60, 0.09,0.08,"e(k)",  11,0 );
       rtbartextworld(0.89,0.60, 0.08,0.06,"LAG",   11,0 );
       rtbartextworld(0.45,0.45, 0.10,0.10,"F(x)",  11,0 );
       rtbartextworld(0.30,0.28, 0.09,0.08,"e(k)",  11,0 );
       rtbartextworld(0.45,0.28, 0.10,0.10,"F(x)1", 11,0 );
       rtbartextworld(0.62,0.20, 0.10,0.10,"e",     11,0 );
       rtbartextworld(0.45,0.12, 0.10,0.10,"F(x)2", 11,0 );


 setcolor(15);
       /* LINES OF BLOCK DIAGRAM */

 rtmoveworldabs(0.02,0.84); rtlineworldabs(0.04,0.84);

 rtmoveworldabs(0.09,0.84); rtlineworldabs(0.13,0.84);

 rtmoveworldabs(0.19,0.84); rtlineworldabs(0.23,0.84);

 rtmoveworldabs(0.29,0.84); rtlineworldabs(0.31,0.84);

 rtmoveworldabs(0.31,0.84); rtlineworldabs(0.31,0.77);

 rtmoveworldabs(0.31,0.77); rtlineworldabs(0.33,0.77);

 rtmoveworldabs(0.41,0.75); rtlineworldabs(0.45,0.75);

 rtmoveworldabs(0.51,0.75); rtlineworldabs(0.55,0.75);

 rtmoveworldabs(0.61,0.75); rtlineworldabs(0.63,0.75);

 rtmoveworldabs(0.63,0.75); rtlineworldabs(0.63,0.67);

 rtmoveworldabs(0.63,0.67); rtlineworldabs(0.65,0.67);

 rtmoveworldabs(0.73,0.67); rtlineworldabs(0.77,0.67);

 rtmoveworldabs(0.85,0.65); rtlineworldabs(0.89,0.65);

 rtmoveworldabs(0.95,0.65); rtlineworldabs(0.985,0.65);

 rtmoveworldabs(0.40,0.50); rtlineworldabs(0.45,0.50);

 rtmoveworldabs(0.55,0.50); rtlineworldabs(0.63,0.50);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารนี้ทุกครั้งที่มีการนำไปใช้

```
rtmoveworldabs(0.63,0.50); rtlineworldabs(0.63,0.62);
rtmoveworldabs(0.63,0.62); rtlineworldabs(0.65,0.62);
rtmoveworldabs(0.38,0.33); rtlineworldabs(0.45,0.33);
rtmoveworldabs(0.55,0.33); rtlineworldabs(0.58,0.33);
rtmoveworldabs(0.55,0.18); rtlineworldabs(0.58,0.18);
rtmoveworldabs(0.58,0.33); rtlineworldabs(0.58,0.28);
rtmoveworldabs(0.58,0.18); rtlineworldabs(0.58,0.23);
rtmoveworldabs(0.58,0.23); rtlineworldabs(0.62,0.23);
rtmoveworldabs(0.58,0.28); rtlineworldabs(0.62,0.28);
rtmoveworldabs(0.72,0.255); rtlineworldabs(0.75,0.255);
rtmoveworldabs(0.75,0.255); rtlineworldabs(0.75,0.62);
rtmoveworldabs(0.75,0.62); rtlineworldabs(0.77,0.62);
rtmoveworldabs(0.40,0.17); rtlineworldabs(0.45,0.17);
rtmoveworldabs(0.40,0.17); rtlineworldabs(0.40,0.33);
rtmoveworldabs(0.25,0.35); rtlineworldabs(0.30,0.35);
rtmoveworldabs(0.25,0.30); rtlineworldabs(0.30,0.30);
rtmoveworldabs(0.25,0.30); rtlineworldabs(0.25,0.25);
rtmoveworldabs(0.25,0.35); rtlineworldabs(0.25,0.40);
rtmoveworldabs(0.15,0.73); rtlineworldabs(0.15,0.07);
rtmoveworldabs(0.15,0.73); rtlineworldabs(0.33,0.73);
rtmoveworldabs(0.15,0.07); rtlineworldabs(0.97,0.07);
rtmoveworldabs(0.97,0.07); rtlineworldabs(0.97,0.65);

   /* SIGNAL NAMES */
rtmoveworldabs(0.001,0.89); outtext("<MS>");
rtmoveworldabs(0.30,0.85); outtext("<MS>");
rtmoveworldabs(0.50,0.82); outtext("<Ts>");
rtmoveworldabs(0.85,0.72); outtext("<T>=<PV>");
rtmoveworldabs(0.37,0.51); outtext("1");
rtmoveworldabs(0.18,0.40); outtext("Setpoint=T");
rtmoveworldabs(0.23,0.20); outtext("Ti");
rtmoveworldabs(0.18,0.16); outtext("Temp feed");
rtmoveworldabs(0.20,0.12); outtext("liquid");
rtmoveworldabs(0.17,0.68); outtext("feedback");

setcolor(9);
rtmoveworldabs(0.38,0.92); outtext("Block Diagram");
}

block()
{
  /* SET WINDOW TO CENTER OF SCREEN */
  rtsetpercentwindow(rtstat[3],0.0,0.0,0.99,0.99);
  rtsetwin2plotratio(rtstat[3],0.0,0.0,0.0,0.0);

      drawdiagram();

      /* LIMIT WINDOW */
      rtborderwindow(rtstat[3],7);

      getch();

  rtclosegraphics(rtstat,1);

}

                    /*** interf.c ***/
#include "stdio.h"
#include "dos.h"
#include "string.h"

port_init(port)
int port;
{   /* INITAIAL PORT */
    union REGS r;
    r.x.dx = port;
    r.h.ah = 0;
    r.h.al = 0xe3;   /* 9600 */
    int86(0x14,&r,&r);
    return 0xe3;
}

r_port(port)
int port;
{    /* READ DATA */
    union REGS r;
    r.x.dx = port;
    r.h.ah = 2;
    int86(0x14,&r,&r);
    return (r.h.al);
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

/*** logo.c ***/

```
#include"stdio.h"
#include"conio.h"
#include"graphics.h"

logo()
{
        int x,y,r;
        int driver,mode=VGAHI;
        driver=DETECT;
        clrscr();
        initgraph(&driver,&mode,"");

         setcolor(LIGHTBLUE);
         settextstyle(DEFAULT_FONT,HORIZ_DIR,6);
         x=1,y=1,r=1;
         for(r=1;r<130;x+=1,y+=1,r+=1)
         outtextxy(x,y,"simulator");
         setcolor(LIGHTCYAN);
         outtextxy(x,y,"SIMULATOR");

         setcolor(RED);
         settextstyle(DEFAULT_FONT,HORIZ_DIR,4);
         outtextxy(200,250,"BY MCS-51");

         setcolor(GREEN);
         settextstyle(DEFAULT_FONT,HORIZ_DIR,2);
         outtextxy(130,350,"DEVELOPMENT BY INSTRUMEN-38");

     getch();
     restorecrtmode();
     closegraph();
}
```

/*** main.c ***/

```
#include <stdio.h>

extern prndriver, prnmode, prnport, orient, formfeed;   /* set printer */
extern cport, baud, parity, stopbits, numbits;          /* set serial  */
extern s_setpoint, s_high_a, s_low_a, d_select;        /* set parameter */

main()
{
        logo();
        port_init();
        r_port();
        main_menu();
        clrscr();
}
```

/*** para.c ***/

```
#include "stdio.h"

int  s_setpoint,s_high_a, s_low_a;

extern  s_setpoint, s_high_a, s_low_a;

int set_para();

int set_para()
{
   int  p_choice;
   int  setpoint, high_a, low_a;

   clrscr();
   background();
   while((p_choice = pulldown(4)) != -1)
     {
        switch(p_choice)
         {
              case 0:
                 while((setpoint = pulldown(5)) != -1)
                   {
                    switch(setpoint)
                      {
                        case 0:
                                s_setpoint = 0;
                                break;
                        case 1:
                                s_setpoint = 10;
                                break;
                        case 2:
                                s_setpoint = 15;
                                break;
                        case 3:
                                s_setpoint = 20;
                                break;
                        case 4:
                                s_setpoint = 25;
                                break;
                        case 5:
                                s_setpoint = 30;
                                break;
                        case 6:
                                s_setpoint = 35;
                                break;
                        case 7:
                                s_setpoint = 40;
                                break;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                case 8:
                        s_setpoint = 45;
                        break;
                case 9:
                        s_setpoint = 50;
                        break;
                case 10:
                        s_setpoint = 55;
                        break;
                case 11:
                        s_setpoint = 60;
                        break;
                case 12:
                        s_setpoint = 65;
                        break;
                case 13:
                        s_setpoint = 70;
                        break;
                case 14:
                        s_setpoint = 75;
                        break;
                case 15:
                        s_setpoint = 80;
                        break;
                case 16:
                        s_setpoint = 85;
                        break;
                case 17:
                        s_setpoint = 90;
                        break;
                case 18:
                        s_setpoint = 95;
                        break;
                case 19:
                        s_setpoint = 100;
                        break;
            }
            break;
        }
        restore_video(5);
        break;
case 1:
    while((high_a = pulldown(6)) != -1)
        {
        switch(high_a)
          {
            case 0:
                    s_high_a = 0;
                    break;
            case 1:
                    s_high_a = 10;
                    break;
            case 2:
                    s_high_a = 15;
                    break;
            case 3:
                    s_high_a = 20;
                    break;
            case 4:
                    s_high_a = 25;
                    break;
            case 5:
                    s_high_a = 30;
                    break;
            case 6:
                    s_high_a = 35;
                    break;
            case 7:
                    s_high_a = 40;
                    break;
            case 8:
                    s_high_a = 45;
                    break;
            case 9:
                    s_high_a = 50;
                    break;
            case 10:
                    s_high_a = 55;
                    break;
            case 11:
                    s_high_a = 60;
                    break;
            case 12:
                    s_high_a = 65;
                    break;
            case 13:
                    s_high_a = 70;
                    break;
            case 14:
                    s_high_a = 75;
                    break;
            case 15:
                    s_high_a = 80;
                    break;
            case 16:
                    s_high_a = 85;
                    break;
            case 17:
                    s_high_a = 90;
                    break;
            case 18:
                    s_high_a = 95;
                    break;
            case 19:
                    s_high_a = 100;
                    break;
          }
          break;
        }
    restore_video(6);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
case 2:
    while((low_a = pulldown(7)) != -1)
      {
       switch(low_a)
        {

            case 0:
                    s_low_a = 0;
                    break;
            case 1:
                    s_low_a = 10;
                    break;
            case 2:
                    s_low_a = 15;
                    break;
            case 3:
                    s_low_a = 20;
                    break;
            case 4:
                    s_low_a = 25;
                    break;
            case 5:
                    s_low_a = 30;
                    break;
            case 6:
                    s_low_a = 35;
                    break;
            case 7:
                    s_low_a = 40;
                    break;
            case 8:
                    s_low_a = 45;
                   break;
           case 9:
                   s_low_a = 50;
                   break;
           case 10:
                   s_low_a = 55;
                   break;
           case 11:
                   s_low_a = 60;
                   break;
           case 12:
                   s_low_a = 65;
                   break;
           case 13:
                   s_low_a = 70;
                   break;
           case 14:
                   s_low_a = 75;
                   break;
           case 15:
                   s_low_a = 80;
                   break;
           case 16:
                   s_low_a = 85;
                   break;
           case 17:
                   s_low_a = 90;
                   break;
           case 18:
                   s_low_a = 95;
                   break;
           case 19:
                   s_low_a = 100;
                   break;

                }
                break;
            }
            restore_video(7);
            break;
                }
            }
            restore_video(4);
        }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                /*** port.c ***/
#include "stdio.h"
#include "conio.h"

int prndriver,prnmode,prnport,orient,formfeed;
int  cport,baud,parity,stopbits,numbits;

extern  prndriver,prnmode,prnport,orientfeed;    /* set printer */
extern  cport,baud,parity,stopbits,numbits;      /* set port */

int printer();
int serial();

int printer()
{
   char ch;
   int i;

      clrscr();
      gotoxy(1,2);
      textcolor(15);
        for (i=0;i<=79;i++)   cprintf("*");
      textcolor(BLUE );
        for (i=0;i<=1759;i++) cprintf("*");
      textcolor(15);
        for (i=0;i<=79;i++)   cprintf("*");

      textcolor(YELLOW);
      textbackground(BLUE);
      gotoxy(10,5);
      cprintf(" STATUS OF PRINTER    PORT : LPT1      DEFAULT : EPSON 9 PIN ");
      gotoxy(10,6);
      cprintf("                      PAGE : PORTRAIT  ");
      textcolor(GREEN);

      gotoxy(9,9);    cprintf("SET DRIVER MODE        SET PORT        SET PAGE");
      gotoxy(10,10);    cprintf(") EPSON 9 PIN          ) LPT1          ) PORTRAIT ");
      gotoxy(10,11);    cprintf(") EPSON 24 PIN         ) LPT2          ) LANDSCAPE");
      gotoxy(10,12);    cprintf(") HP LASERJET          ) COM1 ");
      gotoxy(10,13);    cprintf(") HP DESKJET B&W       ) COM2 ");
      gotoxy(10,14);    cprintf(") HP PAINTJET COL");



      textcolor(RED);
      gotoxy(8,10);  cprintf(" 1");
      gotoxy(8,11);  cprintf(" 2");
      gotoxy(8,12);  cprintf(" 3");
      gotoxy(8,13);  cprintf(" 4");
      gotoxy(8,14);  cprintf(" 5");
      gotoxy(32,10); cprintf("F1");
      gotoxy(32,11); cprintf("F2");
      gotoxy(32,12); cprintf("F3");
      gotoxy(32,13); cprintf("F4");
      gotoxy(51,10); cprintf("F5");
      gotoxy(51,11); cprintf("F6");

      gotoxy(6,22);
      cprintf("SELECT THE PARAMETER OF PRINTER ");

      textcolor(YELLOW);

      if(prnmode == 0)  { gotoxy(60,5);  cprintf("EPSON 9 PIN     ");
      if(prnmode == 3)  { gotoxy(60,5);  cprintf("EPSON 24 PIN    ");
      if(prnmode == 7)  { gotoxy(60,5);  cprintf("HP LASERJET     ");
      if(prnmode == 7)  { gotoxy(60,5);  cprintf("HP DESKJET B&W  ");
      if(prnmode == 15) { gotoxy(60,5);  cprintf("HP PAINTJET COL ");
      if(prnport == 0)  { gotoxy(39,5);  cprintf("LPT1");   }
      if(prnport == 1)  { gotoxy(39,5);  cprintf("LPT2");   }
      if(prnport == 2)  { gotoxy(39,5);  cprintf("COM1");   }
      if(prnport == 3)  { gotoxy(39,5);  cprintf("COM2");   }
      if(orient == 0)   { gotoxy(39,6);  cprintf("PORTRAIT  "); }
      if(orient == 1)   { gotoxy(39,6);  cprintf("LANDSCAPE "); }

      textcolor(YELLOW);
      do{
        ch = getch();
          switch(ch){

      /* set MODE*/
                  case '1': prnmode = 0;
                            gotoxy(60,5);
                            cprintf("EPSON 9 PIN     ");
                            break;
                  case '2': prnmode = 3;
                            gotoxy(60,5);
                            cprintf("EPSON 24 PIN    ");
                            break;
                  case '3': prnmode = 7;
                            gotoxy(60,5);
                            cprintf("HP LASERJET     ");
                            break;
                  case '4': prnmode = 7;
                            gotoxy(60,5);
                            cprintf("HP DESKJET B&W  ");
                            break;
                  case '5': prnmode = 15;
                            gotoxy(60,5);
                            cprintf("HP PAINTJET COL ");
                            break;

      /*set PORT*/  case 59 : prnport = 0;
                            gotoxy(39,5);
                            cprintf("LPT1");
                            break;
                  case 60 : prnport = 1;
                            gotoxy(39,5);
                            cprintf("LPT2");
                            break;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                case 61 : prnport = 2;
                          gotoxy(39,5);
                          cprintf("COM1");
                          break;
                case 62 : prnport = 3;
                          gotoxy(39,5);
                          cprintf("COM2");
                          break;
/*set PAGE*/
                case 63 : orient = 0;
                          gotoxy(39,6);
                          cprintf("PORTRAIT  ");
                          break;
                case 64 : orient = 1;
                          gotoxy(39,6);
                          cprintf("LANDSCAPE ");
                          break;
                      }
                  prndriver = -16;  /* set DRIVER */
                  formfeed  =   0;  /* set FORMFEED */

              }while(ch != 27);
          textbackground(BLACK);
          textcolor(WHITE);
       clrscr();
    }

    int  serial()
    {
       char ch;.
       int i;

           clrscr();
           gotoxy(1,2);
           textcolor(15);
             for (i=0;i<=79;i++)   cprintf("=");
           textcolor(BLUE );
             for (i=0;i<=1759;i++) cprintf("=");
           textcolor(15);
             for (i=0;i<=79;i++)   cprintf("=");

           textcolor(YELLOW);
           textbackground(BLUE);
           gotoxy(5,5);
           cprintf(" STATUS OF SERIAL PORT  COM1 : 9600 :  NO PARITY : 1 STOPBIT : 8 DATA");
           textcolor(GREEN);

           gotoxy(5,9);    cprintf("SET BAUD RATE    SET COM PORT   SET PARITY      SET STOPBIT    SET DATABIT");
           gotoxy(6,10);   cprintf(")  110  Bd.       ) COM1        ) NO PARITY      ) 1 BIT        ) 7 BIT");
           gotoxy(6,11);   cprintf(")  150  Bd.       ) COM2        )ODD PARITY      ) 2 BIT        ) 8 BIT");
           gotoxy(6,12);   cprintf(")  300  Bd.                     )EVEN RARITY");
           gotoxy(6,13);   cprintf(")  600  Bd.");
           gotoxy(6,14);   cprintf(") 1200  Bd.");
           gotoxy(6,15);   cprintf(") 2400  Bd.");
           gotoxy(6,16);   cprintf(") 4800  Bd.");
           gotoxy(6,17);   cprintf(") 9600  Bd.");
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/*set PARITY*/
                case 'N' : parity = 0;
                           gotoxy(43,5);
                           cprintf(" NO PARITY ");
                           break;
                case 'n' : parity = 0;
                           gotoxy(43,5);
                           cprintf(" NO PARITY ");
                           break;
                case 'O' : parity = 1;
                           gotoxy(43,5);
                           cprintf("ODD PARITY ");
                           break;
                case 'o' : parity = 1;
                           gotoxy(43,5);
                           cprintf("ODD PARITY ");
                           break;
                case 'E' : parity = 3;
                           gotoxy(43,5);
                           cprintf("EVEN PARITY");
                           break;
                case 'e' : parity = 3;
                           gotoxy(43,5);
                           cprintf("EVEN PARITY");
                           break;
/*set STOPBIT*/
                case 61  : stopbits = 1;
                           gotoxy(56,5);
                           cprintf("1 STOPBIT");
                           break;
                case 62  : stopbits = 2;
                           gotoxy(56,5);
                           cprintf("2 STOPBIT");
                           break;
/*set DATA BIT */
                case 63  : numbits = 7;
                           gotoxy(68,5);
                           cprintf("7 DATA");
                           break;
                case 64  : numbits = 8;
                           gotoxy(68,5);
                           cprintf("8 DATA");
                           break;
                         }
        }while(ch != 27);

    textbackground(BLACK);
    textcolor(WHITE);
  clrscr();
}
```

```
                                                    /*** sub.c ***/
#include"stdio.h"
#include"conio.h"
#include"dos.h"
#include"stdlib.h"

#define BORDER     1
#define ESC       27
#define MAX_FRAME 50
#define REV_VID   113
#define NORM_VID  12

int    d_select;
extern prndriver, prnmode, prnport, orient, formfeed;   /* set printer */
extern cport, baud, parity, stopbits, numbits;          /* set serial  */
extern s_setpoint, s_high_a, s_low_a, d_select;         /* set parameter */

void save_video(),restore_video(),pd_driver();
void goto_xy(),cls(),write_video(),write_string();
void display_menu(),draw_border(),write_char();
char far *vid_mem;

void background();
int main_menu();

struct menu_frame{
  int startx,starty,endx,endy;
  unsigned char *p;
  char **menu;
  char *keys;
  int border,count;
  int active;
  } frame[MAX_FRAME];

  char *main1[] = {
    " SET PARAMETER OF SERIAL PORT     ",
    " SET PARAMETER OF PRINTER         ",
    " TO MENU CONTROL PROGRAM          ",
    " Esc TO QUIT                      ",
    };

  char *main2[] = {
    "    TEMPERATURE SIMULATOR    ",
    "    LEVEL SIMULATOR (FUTURE) ",
    "    FLOW  SIMULATOR (FUTURE) ",
    };

  char *tempn[] = {
    "   SET PARAMETER OF TEMPERATURE SIMULATOR    ",
    "   SELECT DISPLAY MONITOR                    ",
    };

  char *sub_disp[] = {
    "   DISPLAY ON LINE GRAPH                     ",
    "   DISPLAY BLOCK DIAGRAM                     ",
    };
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    "   SET HIGHT ALARM                 ",
    "   SET LOW ALARM                   ",
    };

char *setpoint[] = {
    "     0 %  ",
    "    10 %  ",
    "    15 %  ",
    "    20 %  ",
    "    25 %  ",
    "    30 %  ",
    "    35 %  ",
    "    40 %  ",
    "    45 %  ",
    "    50 %  ",
    "    55 %  ",
    "    60 %  ",
    "    65 %  ",
    "    70 %  ",
    "    75 %  ",
    "    80 %  ",
    "    85 %  ",
    "    90 %  ",
    "    95 %  ",
    "   100 %  ",
    };

 char *high_a[] = {
    "     0 %  ",
    "    10 %  ",
    "    15 %  ",
    "    20 %  ",
    "    25 %  ",
    "    30 %  ",
    "    35 %  ",
    "    40 %  ",
    "    45 %  ",
    "    50 %  ",
    "    55 %  ",
    "    60 %  ",
    "    65 %  ",
    "    70 %  ",
    "    75 %  ",
    "    80 %  ",
    "    85 %  ",
    "    90 %  ",
    "    95 %  ",
    "   100 %  ",
    };

 char *low_a[] = {
    "     0 %  ",
    "    10 %  ",
    "    15 %  ",
    "    20 %  ",
    "    25 %  ",
    "    30 %  ",
    "    35 %  ",
    "    40 %  ",
    "    45 %  ",
    "    50 %  ",
    "    55 %  ",
    "    60 %  ",
    "    65 %  ",
    "    70 %  ",
    "    75 %  ",
    "    80 %  ",
    "    85 %  ",
    "    90 %  ",
    "    95 %  ",
    "   100 %  ",
    };

int main_menu()
{
      int i;
      clrscr();
      cls();
      goto_xy(0,0);
      background();
      getch();

       make_menu(0,main1,"MAIN1",4,5,5,BORDER);
       make_menu(1,main2,"MAIN2",3,20,10,BORDER);
       make_menu(2,tempn,"TEMPN",2,5,18,BORDER);
       make_menu(3,sub_disp,"SUB_DISP",3,5,5,BORDER);
       make_menu(4,parameter,"PARAMETER",3,5,5,BORDER);
       make_menu(5,setpoint,"SETPOINT",20,40,1,BORDER);
       make_menu(6,high_a,"HIGH ALARM",20,40,1,BORDER);
       make_menu(7,low_a,"LOW ALARM",20,40,1,BORDER);
     pd_driver();

}

void pd_driver()
{

   int choice1,choice2,choice3,choice4,
     selection;


   while((choice1 = pulldown(0)) != -1)    /* main1 */
   {
     switch(choice1)
     {

        case 0:
                serial();
                background();
                break;
        case 1:
                printer();
                background();
                break;
        case 2:
          while((choice2 = pulldown(1)) != -1)   /*main2*/
            {
              switch(choice2)
                {

                   case 0:
                       while((choice3 = pulldown(2)) != -1)  /*temp
                         {
                           switch(choice3)
                             {
                           case 0:
                                   set_para();
                                   break;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    char *keys;
    int  count;
    int  x,y;
    int  border;
{
       register int i,len;
       int endx,endy,chioce,vmode;
       unsigned char *p;

        if (num>MAX_FRAME)
        {
           printf("Too many menu !\n");
           return 0;
         }
        if ((y>24)||(y<0)||(x<0)||(x>79))
        {
           printf("Rang error !\n");
           return 0;
         }
       len = 0;

          for(i=0;i<count;i++)
          if(len<strlen(menu[i]))
                len = strlen(menu[i]);
             endx = x + len +1;
             endy = count + 1 + y;
             if ((endy>25)||(endx>80))
              {
                  printf("Menu won't fit \n");
                  return 0;
               }
          p = (unsigned char*) malloc(2*(endx-x+1)*(endy-y+1));
            if(!p)
                exit(1);

        frame[num].startx = x;
        frame[num].starty = y;
        frame[num].endx = endx;
        frame[num].endy = endy;
        frame[num].p = p;
        frame[num].menu = (char**) menu;
        frame[num].border = border;
        frame[num].keys = keys;
        frame[num].count = count;
        frame[num].active = 0;
           return 1;
}

void display_menu(num)
   int num;
{
      register int y,j;
      char **m;
      y = frame[num].starty + 1;
      m = frame[num].menu;
      for(j = 0;j < frame[num].count;j++,y++)
      write_string(frame[num].startx+1,y,m[j],NORM_VID);
}

void draw_border(num)
   int num;
{
      register int i;
      int endx,endy;
      endx = frame[num].endx;
      endy = frame[num].endy;
      for(i=frame[num].startx;i<=endx;i++)
      {
         write_char(i,frame[num].starty,0xCD,NORM_VID);
         write_char(i,endy,0xCD,NORM_VID);
      }
      for(i=frame[num].starty;i<=endy;i++)
      {
         write_char(frame[num].startx,i,0xBA,NORM_VID);
         write_char(endx,i,0xBA,NORM_VID);
      }
      write_char(frame[num].startx,frame[num].starty,0xC9,NORM_VID);
      write_char(endx,frame[num].starty,0xBB,NORM_VID);
      write_char(frame[num].startx,endy,0xC8,NORM_VID);
      write_char(endx,endy,0xBC,NORM_VID);
}

int pulldown(num)
   int num;
{
      int vmode,choice;

      vmode = video_mode();
      if((vmode!=2) && (vmode!=3) && (vmode!=7))
      {
         printf("video must in 80 colum text mode");
         exit(1);
       }
        if(vmode == 7) vid_mem = (char far *) 0xB0000000;
          else vid_mem = (char far *) 0xB8000000;
        if(!frame[num].active)
         {
           save_video(num);
           frame[num].active = 1;
          }

       if (frame[num].border) draw_border(num);

       display_menu(num);
       return get_resp(num);
}

int get_resp(num)
   int num;
{
      int c,i,j,k,old,row,col,d_select,ret_val;
      char **m,*keys;

      m = frame[num].menu;
      keys = frame[num].keys;
      d_select = 0;
      old = 0;
      while(1)
      {
         row = frame[num].starty + 1 + d_select;
         col = frame[num].startx + 1;
         write_string(col,row,m[d_select],REV_VID);
         if(old != d_select)
         {
            row = frame[num].starty + 1 + old;
            write_string(col,row,m[old],NORM_VID);
            old = d_select;
         }
         c = get_key();
         if(c == 0)
         {
            c = get_key();
            switch(c)
            {
              case 0:
                      d_select = 2;
                      display();
                      background();
                      break;
              case 1:
                      d_select = 1;
                      display();
                      background();
                      break;
              }
            }
           restore_video(3);
           break;

              }
           }
           restore_video(2);
           break;

        case 1:
           break;
        case 2:
           break;
              }
           }

           restore_video(1);
           break;

        case 3:
           exit(0);
           break;
      }
      }
    restore_video(0);
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        t = v;
        for(i = frame[num].startx+1;i<frame[num].endx; i++)
         {
            v += frame[num].starty*160 +(i*2);
            *v++ = 205;
            *v = NORM_VID;
            v = t;
            v += frame[num].endy*160 + (i*2);
            *v++  = 205;
            *v = NORM_VID;
            v = t;
         }
         for( i = frame[num].starty+1; i<frame[num].endy; i++)
         {
            v += i*160 + (frame[num].startx*2);
            *v++ = 179;
            *v = NORM_VID;
            v = t;
            v += (i * 160) + frame[num].endx*2;
            *v++  = 179;
            *v = NORM_VID;
            v = t;
          }
       write_char(frame[num].startx,frame[num].starty,213,NORM_VID);
       write_char(frame[num].startx,frame[num].endy,212,NORM_VID);
       write_char(frame[num].endx,frame[num].starty,184,NORM_VID);
       write_char(frame[num].endx,frame[num].endy,190,NORM_VID);
}

get_resp(num)
   int num;
{
     union inkey {
         char ch[2];
         int i;
      } c;
    int arrow_chioce = 0, key_chioce;
    int x, y;
    x = frame[num].startx+1;
    y = frame[num].starty+1;

        write_string(x,y,frame[num].menu[0],REV_VID);
           for(;;)
           {
              while(!bioskey(1));
              c.i = bioskey(0);

              write_string(x,y+arrow_chioce,
              frame[num].menu[arrow_chioce],NORM_VID);

            if(c.ch[0])
              {
                 key_chioce = is_in(frame[num].keys,tolower(c.ch[0]));
                 if(key_chioce) return key_chioce -1;
                 switch(c.ch[0])
                 {
                      case '\r' : return arrow_chioce;
                      case ' ' : arrow_chioce++;
                                 break;
                      case ESC : return -1;
                  }
              }
            else
              {
                 switch(c.ch[1])
                 {
                      case 72 : arrow_chioce--;
                                break;
                      case 80 : arrow_chioce++;
                                break;
                 }
              }
            if(arrow_chioce == frame[num].count) arrow_chioce = 0
            if(arrow_chioce < 0) arrow_chioce = frame[num].count-

            write_string(x,y+arrow_chioce,
            frame[num].menu[arrow_chioce],REV_VID);

           }
}

void write_string(x,y,p,attrib)
   int x, y;
   char *p;
   int  attrib;
{
   register int i;
   char far *v;

   v = vid_mem;
   v += y*160 + x*2;
   for(i = x; *p; i++)
   {
      *v++ = *p++;
      *v++ = attrib;
   }
}

void write_char(x,y,ch,attrib)
{
   register int i;
   char far *v;

   v = vid_mem;
   v += y*160 + x*2;
   *v++ = ch;
   *v = attrib;
}

void save_video(num)
   int num;
{
    register int i, j;
    char *buf_ptr;
    char far *v, far *t;

    buf_ptr = frame[num].p;
    v = vid_mem;
    for(i = frame[num].starty; i<frame[num].endy+1; i++)
        for(j = frame[num].startx; j<frame[num].endx+1; j++)
        {
             t = v + (i*160) + (j*2);
             *buf_ptr++ = *t++;
             *buf_ptr++ = *t;
             *(t-1) = ' ';
        }
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    char far *v, far *t;
    char *buf_ptr;

    buf_ptr = frame[num].p;
    v = vid_mem;
    t = v;
       for(i = frame[num].starty; i<frame[num].endy+1; i++)
          for(j = frame[num].startx; j<frame[num].endx+1; j++)
          {
             v = t;
             v += (i*160) + (j*2);
             *v++ = *buf_ptr++;
             *v = *buf_ptr++;
          }
       frame[num].active = 0;
}

void cls()
{
       union REGS r;

       r.h.ah = 6;
       r.h.al = 0;
       r.h.ch = 0;
       r.h.cl = 0;
       r.h.dh = 14;
       r.h.dl = 79;
       r.h.bh = 7;
    int86(0x10, &r, &r);
}

void goto_xy(x,y)
   int x, y;
{
       union REGS r;
       r.h.ah = 2;
       r.h.dl = x;
       r.h.dh = y;
       r.h.bl = 0;
    int86(0x10, &r, &r);
}

video_mode()
{
    union REGS r;

    r.h.ah = 15;
    return int86(0x10, &r, &r) & 255;
}


is_in(s,c)
  char *s,c;
{
    register int i;
    for(i = 0; *s; i++)
       if (*s++ ==c) return i+1;
    return 0;
}
```

```
gotoxy(1,2);
textcolor(15);
   for (i=0;i<=79;i++)   cprintf("=");
textcolor(LIGHTBLUE );
   for (i=0;i<=1839;i++)  cprintf("=");
textcolor(15);
   for (i=0;i<=78;i++)   cprintf("=");
textcolor(WHITE);
gotoxy(13,6);
cprintf("                                                    ");
gotoxy(13,7);
cprintf("  KING MONGKUT 'S INSTITUTE OF TECHNOLOGY LADKRABANG  ");
gotoxy(13,8);
cprintf("                                                    ");
gotoxy(13,9);
cprintf("                                                    ");
gotoxy(13,10);
cprintf("                PROCESS  SIMULATOR                  ");
gotoxy(13,11);
cprintf("                   BY MCS-51                        ");
gotoxy(13,12);
cprintf("                                                    ");
gotoxy(13,13);
cprintf("                FACULTY OF ENGINEERING              ");
gotoxy(13,14);
cprintf("                                                    ");
gotoxy(13,15);
cprintf("      DEPARTMENT OF INSTRUMENTATION ENGINEERING     ");
gotoxy(13,16);
cprintf("                                                    ");
gotoxy(13,17);
cprintf("      BY Mr. THAWAN     WORACHIN      36013285      ");
gotoxy(13,18);
cprintf("         Mr. TAWEEKIAT  SODSIRISAWAD  36013286      ");
gotoxy(13,19);
cprintf("         Mr. PHISANURAT MANACON       36013295      ");
gotoxy(13,20);
cprintf("                                                    ");
gotoxy(1,25);
}
```

```
    /*** project.prj ***/
\\tc\\project\\para.c
\\tc\\project\\port.c
\\tc\\project\\main.c
\\tc\\project\\sub.c
\\tc\\project\\disp.c
\\tc\\project\\interf.c
\\tc\\project\\logo.c
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            /*** plant.c ***/
#include<dos.h>
#include<stdio.h>
void mode(),line(),box(),fill_box();
void mempoint(),palette(),xhairs();
void circle(),plot_circle(),fill_circle();
void ellipse(),plot_ellipse(),fill_ellipse();

double asp_ratio;

/*plant()*/
main()
{
  clrscr();
  mode(4);
  palette(0);

  box(80,160,140,200,1); /*TANK*/
  fill_box(90,162,140,199,2);
  line(90,160,90,270,3);line(86,265,90,270,3);line(94,265,90,270,3);

  box(30,70,42,120,1);  /*controller*/
  line(39,53,39,70,1);line(33,53,33,70,1);
  box(31,71,41,120,1);

  circle(69,105,6,1);   /*TY21*/
  line(42,105,62,105,3);line(50,104,50,106,0);line(53,104,53,106,0);
  line(56,104,56,106,0); line(59,104,59,106,0);
  line(62,105,59,102,3);line(62,105,59,108,3);
  line(76,105,89,105,3);
  line(89,105,86,102,3);line(89,105,86,108,3); line(100,105,105,105,2);

  /*valve*/
  line(100,95,110,95,2);line(100,115,110,115,2);
  line(100,95,110,115,2);line(110,95,100,115,2);
  fill_circle(95,105,5,2); fill_box(95,100,103,112,0);line(95,105,105,105,2);

  circle(67,168,6,1);   /*TT21*/
  line(38,168,61,168,3);line(41,167,41,169,0);line(44,167,44,169,0);
  line(47,167,47,169,0);line(50,167,50,169,0);line(53,167,53,169,0);
  line(56,167,56,169,0);line(59,167,59,169,0);
  line(38,121,38,168,3);line(37,165,39,165,0);line(37,160,39,160,0);
  line(37,155,39,155,0);line(37,150,39,150,0);line(37,145,39,145,0);
  line(37,140,39,140,0);line(37,135,39,135,0);
  line(38,121,41,124,3);line(38,121,35,124,3);
  line(74,168,97,168,3);
  circle(98,168,1,3);

  ellipse(67,180,4,4); line(73,180,133,180,3);
  fill_ellipse(67,180,4,1);line(63,176,63,185,3);line(71,176,71,185,3);
  line(107,177,110,185,3); line(109,177,106,185,3);
  line(131,177,134,185,3); line(133,177,130,185,3);
  box(140,130,153,146,1); /*Trap*/
  line(153,138,170,138,3); line(170,138,167,135,3);line(170,138,167,141,3);
  line(105,95,105,53,3);  /*STEAM*/
  line(105,115,105,190,3); circle(108,190,3,3); fill_box(106,187,112,19
  line(111,170,111,190,3); circle(114,170,3,3); fill_box(112,170,118,17
  line(117,170,117,190,3); circle(120,190,3,3); fill_box(118,187,124,19
  line(123,170,123,190,3); circle(126,170,3,3); fill_box(124,170,130,17
  line(129,170,129,190,3); circle(132,190,3,3); fill_box(130,187,135,19

  box(1,1,185,300,2);

  gotoxy(8,3);printf("CONTROLLER");
  gotoxy(2,13);printf("STEAM FLOW");
  gotoxy(2,15);printf("FEED LIQUID");
  gotoxy(19,22);printf("Condensate");
  gotoxy(20,19);printf("Trap");
  gotoxy(27,11);printf("Product");
  gotoxy(3,6);printf("P,I,D");
  gotoxy(15,7);printf("MV");
  gotoxy(19,4);printf("PV");
  gotoxy(6,4);printf("SP");
  gotoxy(15,9);printf("TY");
  gotoxy(15,10);printf("21");
  gotoxy(19,9);printf("TT");
  gotoxy(19,10);printf("21");

   getchar();
   mode(3);
  }

  void palette(pnum)
  int pnum;
  {
   union REGS r;
    r.h.bh = 1;
    r.h.bl = pnum;
   r.h.ah = 11;
    int86(0x10,&r,&r);
   }

void mode(int mode_code)
{
  union REGS r;
    r.h.al = mode_code;
    r.h.ah = 0;
    int86(0x10,&r,&r);
}

void box(int startx,int starty,int endx,int endy,int color_code)
{
  line(startx,starty,endx,starty,color_code);
  line(startx,starty,startx,endy,color_code);
  line(startx,endy,endx,endy,color_code);
  line(endx,starty,endx,endy,color_code);
  }

void line(int startx,int starty,int endx,int endy,int color)
{
  register int t,distance ;
  int x=0,y=0,delta_x,delta_y;
  int incx,incy;

  delta_x = endx - startx;
```

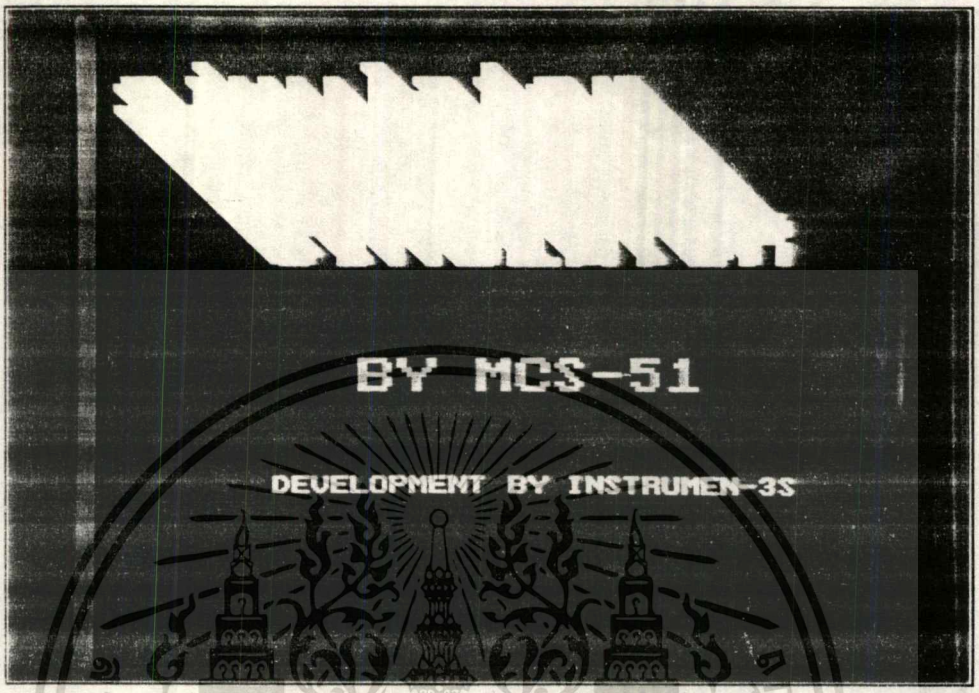
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  if(delta_y > 0) incy = 1;
  else if(delta_y == 0) incy = 0;
  else incy =-1;

  delta_x = abs(delta_x);
  delta_y = abs(delta_y);

  distance = delta_x>delta_y ? delta_x : delta_y;

   for(t=0;t<distance+1;t++){
     mempoint(startx,starty,color);
     x+=delta_x;
     y+=delta_y;
      if(x>distance){
        x-=distance;
        startx+=incx;
        }
      if(y>distance){
        y-=distance;
        starty+=incy;
        }
     }
  }

void fill_box(int startx,int starty,int endx,int endy,int color_code)
{
  register int i,begin,end;
     begin = startx<endx ? startx : endx;
     end = startx>endx ? startx : endx;
   for(i=begin;i<end;i++)
     line(i,starty,i,endy,color_code);
  }

void circle(x_center,y_center,radius,color_code)
int x_center,y_center,radius,color_code;
{
  register int x,y,delta;
  asp_ratio=1.0;

  y=radius;
  delta = 3 - 2 * radius;
  for(x=0;x<y; ){
     plot_circle(x,y,x_center,y_center,color_code);
       if(delta < 0)
         delta += 4*x+6;
       else{
         delta += 4*(x-y)+10;
         y--;
        }
        x++;
        }
     x = y;
    if(y) plot_circle(x,y,x_center,y_center,color_code);
  }

void plot_circle(int x,int y,int x_center,int y_center,int color_code)
{
  int startx,starty,endx,endy,x1,y1;

     starty = y*asp_ratio;
     endy = (y+1)*asp_ratio;
     startx = x*asp_ratio;
     endx = (x+1)*asp_ratio;

  for(x1=startx;x1<endx;++x1){
     mempoint(x1+x_center,y+y_center,color_code);
     mempoint(x1+x_center,y_center-y,color_code);
     mempoint(x_center-x1,y_center-y,color_code);
     mempoint(x_center-x1,y+y_center,color_code);
     }
  for(y1=starty;y1<endy;++y1){
     mempoint(y1+x_center,x+y_center,color_code);
     mempoint(y1+x_center,y_center-x,color_code);
     mempoint(x_center-y1,y_center-x,color_code);
     mempoint(x_center-y1,x+y_center,color_code);
     }
  }

void fill_circle(x,y,r,c)
int x,y,r,c;
{
  while(r){
    circle(x,y,r,c);
    r--;
    }
  }

void ellipse(x_center,y_center,radius,color_code)
int x_center,y_center,radius,color_code;
{
  register int x,y,delta;
  asp_ratio= 2.0;

  y= radius;
  delta = 3 - 2 * radius;
  for(x=0;x<y; ){
    plot_ellipse(x,y,x_center,y_center,color_code);
       if(delta < 0)
         delta += 4*x+6;
       else{
         delta += 4*(x-y)+10;
         y--;
         }
         x++;
         }
       x = y;
      if(y) plot_ellipse(x,y,x_center,y_center,color_code);
  }

void plot_ellipse(int x,int y,int x_center,int y_center,int color_code)
{
  int startx,starty,endx,endy,x1,y1;

     starty = y*asp_ratio;
     endy = (y+1)*asp_ratio;
     startx = x*asp_ratio;
     endx = (x+1)*asp_ratio;

  for(x1=startx;x1<endx;++x1){
     mempoint(x1+x_center,y+y_center,color_code);
     mempoint(x1+x_center,y_center-y,color_code);
     mempoint(x_center-x1,y_center-y,color_code);
     mempoint(x_center-x1,y+y_center,color_code);
     }
  for(y1=starty;y1<endy;++y1){
     mempoint(y1+x_center,x+y_center,color_code);
     mempoint(y1+x_center,y_center-x,color_code);
```

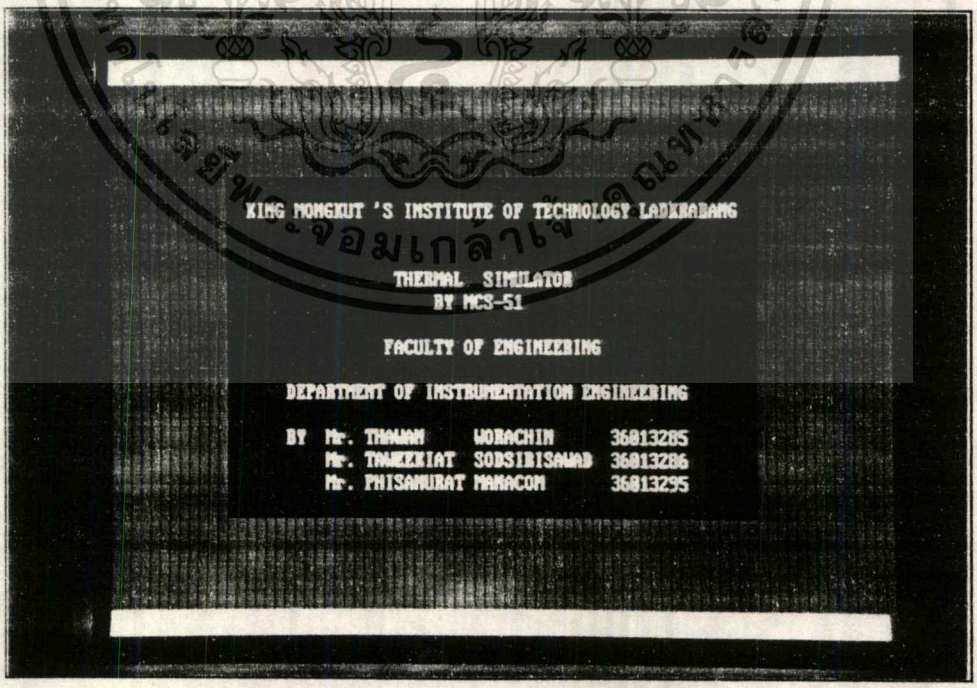
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void fill_ellipse(x,y,r,c)
int x,y,r,c;
{
  while(r){
      ellipse(x,y,r,c);
      r--;
      }
 }

void mempoint(x,y,color_code)
int x,y,color_code;
{
 union mask{
    char c[2];
    int i;
   } bit_mask;
 int i,index,bit_position;
 unsigned char t;
 char xor;
 char far *ptr =(char far *) 0xB8000000;
 bit_mask.i= 0xFF3F;

 if(x<0 || x>199 || y<0 || y>319) return;

 xor=color_code & 128;
 color_code=color_code & 127;

 bit_position = y%4;
 color_code<<=2*(3-bit_position);
 bit_mask.i>>=2*bit_position;

 index = x*40+(y>>2);
   if(x%2) index += 8152;

 if(!xor){
   t = *(ptr+index) & bit_mask.c[0];
   *(ptr+index)= t | color_code;
  }
  else{
   t = *(ptr+index) | (char) 0;
   *(ptr+index)= t | color_code;
   }
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

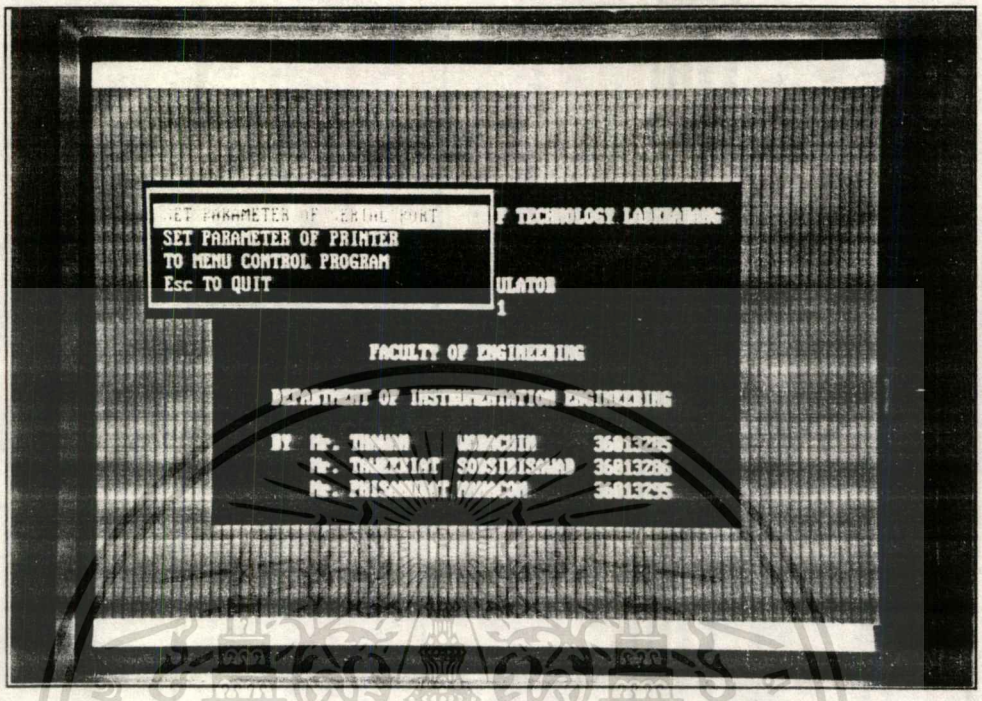


รูปที่ 4.1 แสดงโลโก้ (LOGO) ของเมนู

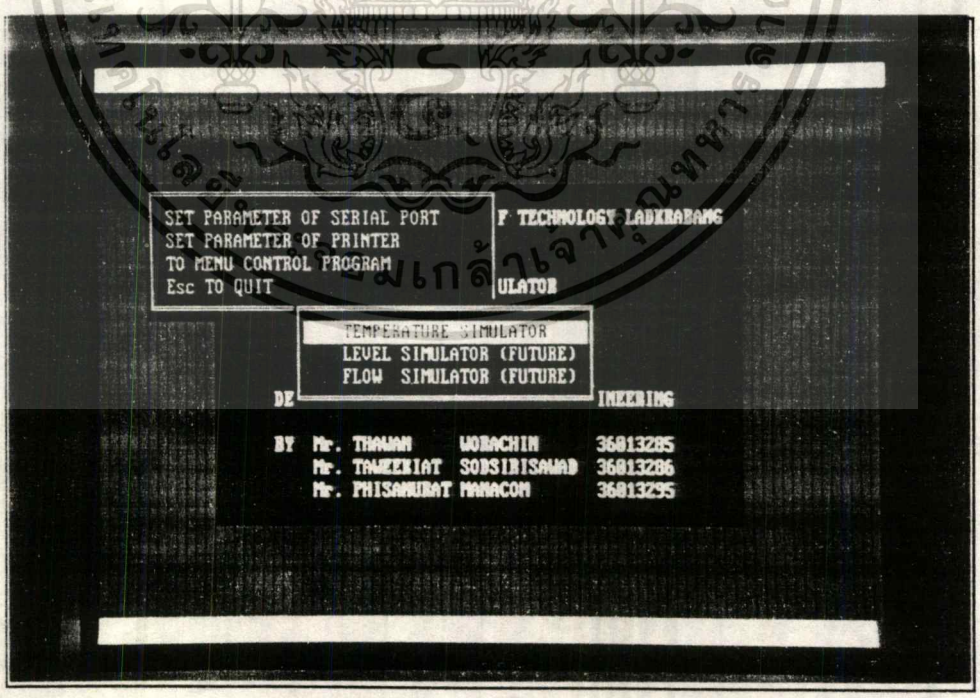


รปที่ 4.2 แสดงเมนู (SHOW TITLE)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

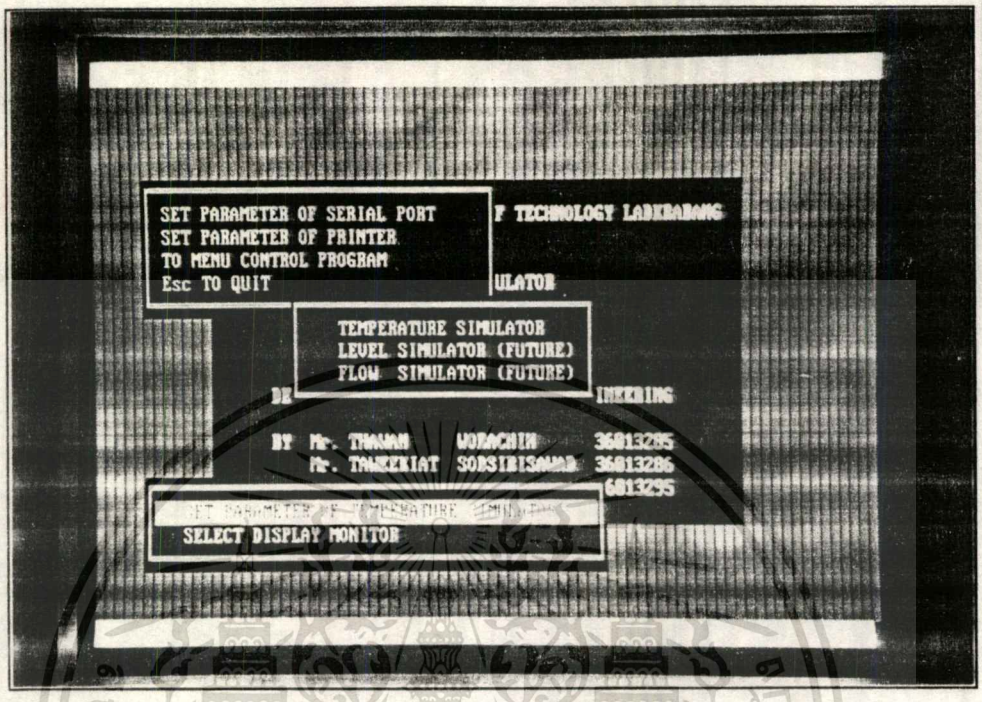


รูปที่ 4.3 แสดงส่วนของเมนูในการเซ็ตตัวแปรของการติดต่อ RS232C

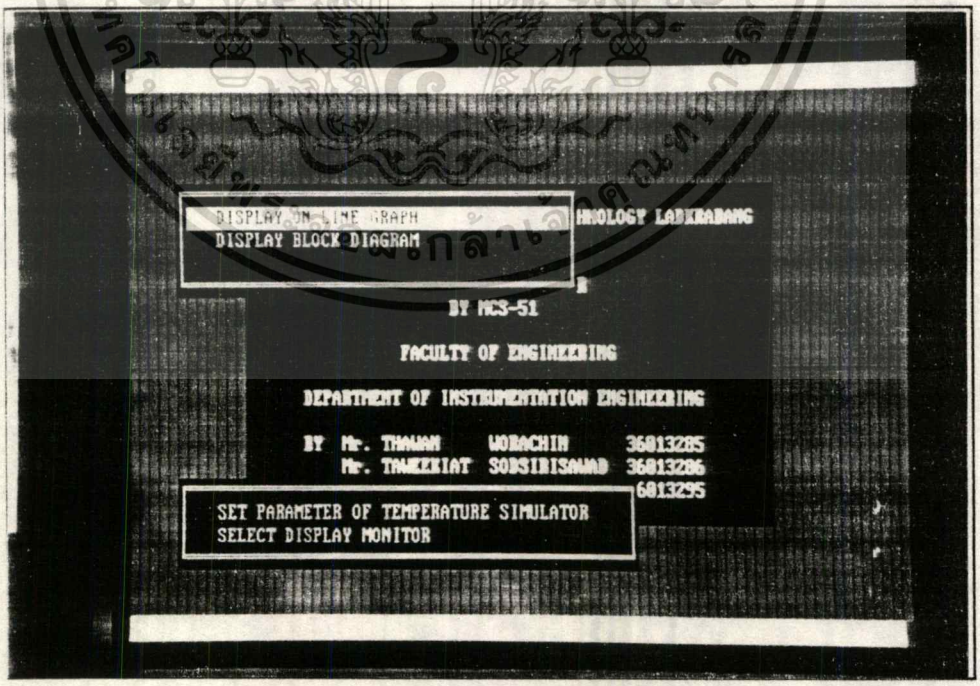


รปที่ 4.4 แสดงเมนูในการเลือก ตัวจำลองกระบวนการที่ใช้ในการจำลอง ซึ่งเลือกใช้การจำลองขบวนการทางความร้อน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



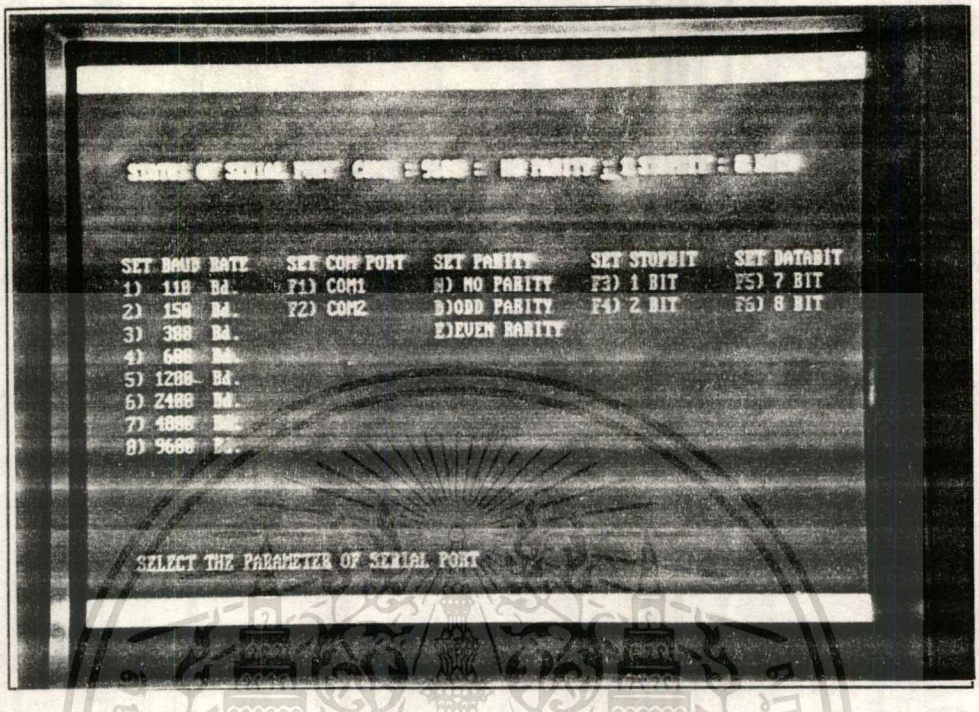
รูปที่ 4.5 แสดงส่วนของเมนูในการเซ็ตตัวแปร เพื่อให้แสดงผลทางหน้าจอ



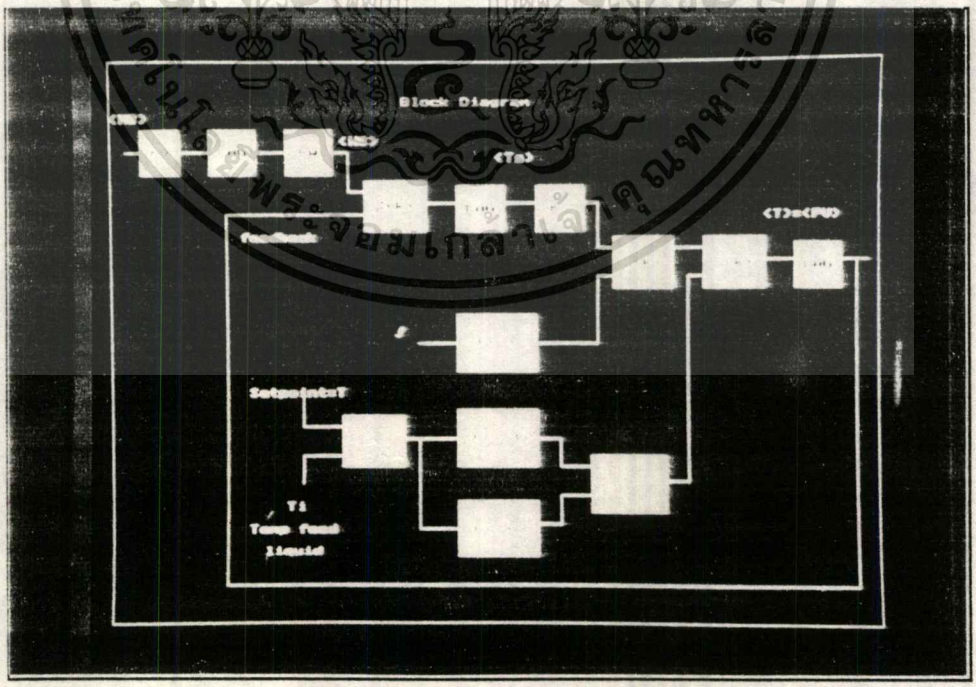
รปที่ 4.6 แสดงเมนูในการเลือก การแสดงผลของเส้นกราฟบนหน้าจอ

หรือเลือกการแสดงผลแบบ บล๊อกไดอะแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

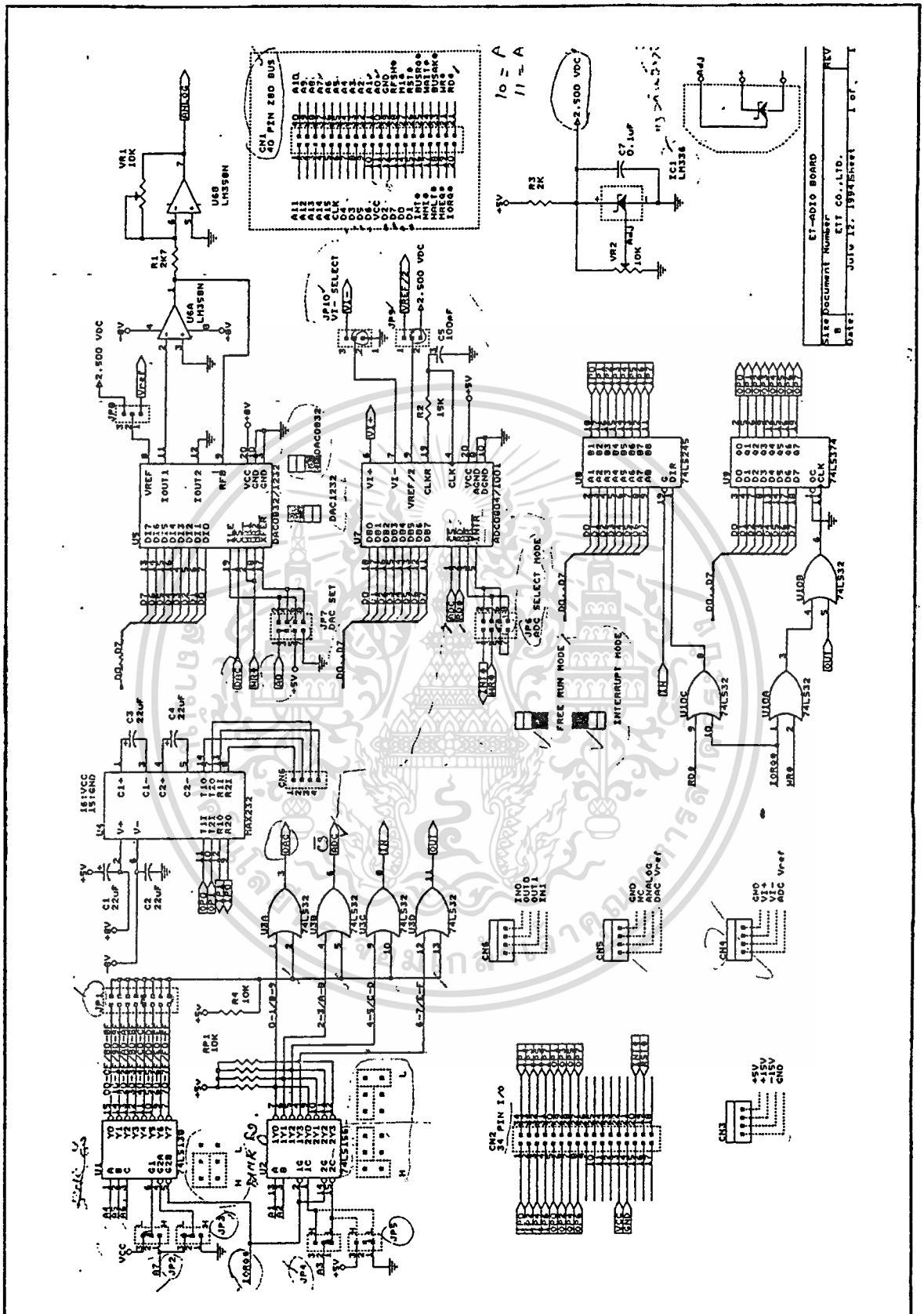


รูปที่ 4.7 แสดงส่วนของเมนูในการเซ็ตตัวแปรที่ใช้ในการติดต่อระหว่างเครื่องคอมพิวเตอร์กับตัวจำลองกระบวนการทางความร้อน

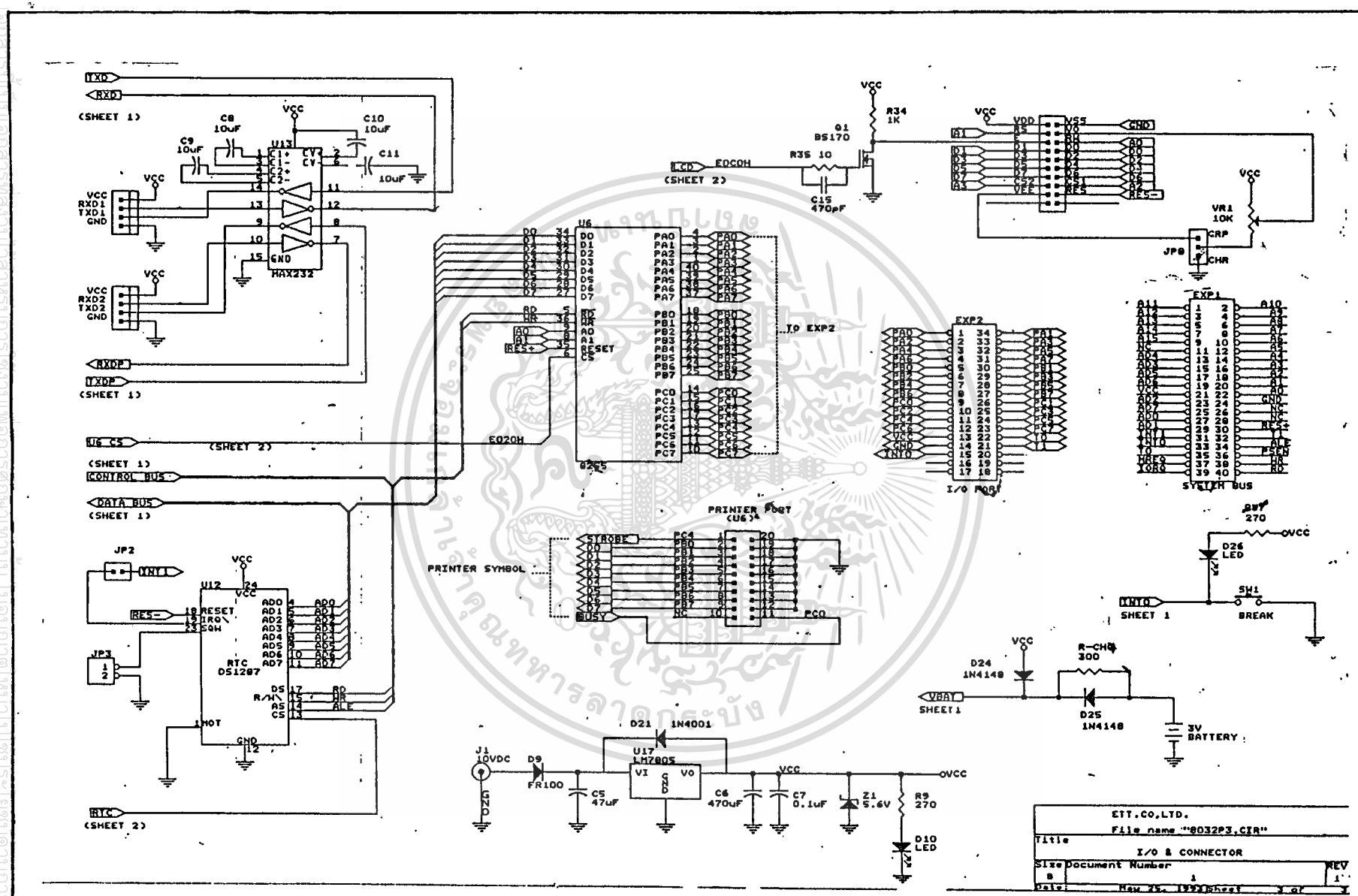


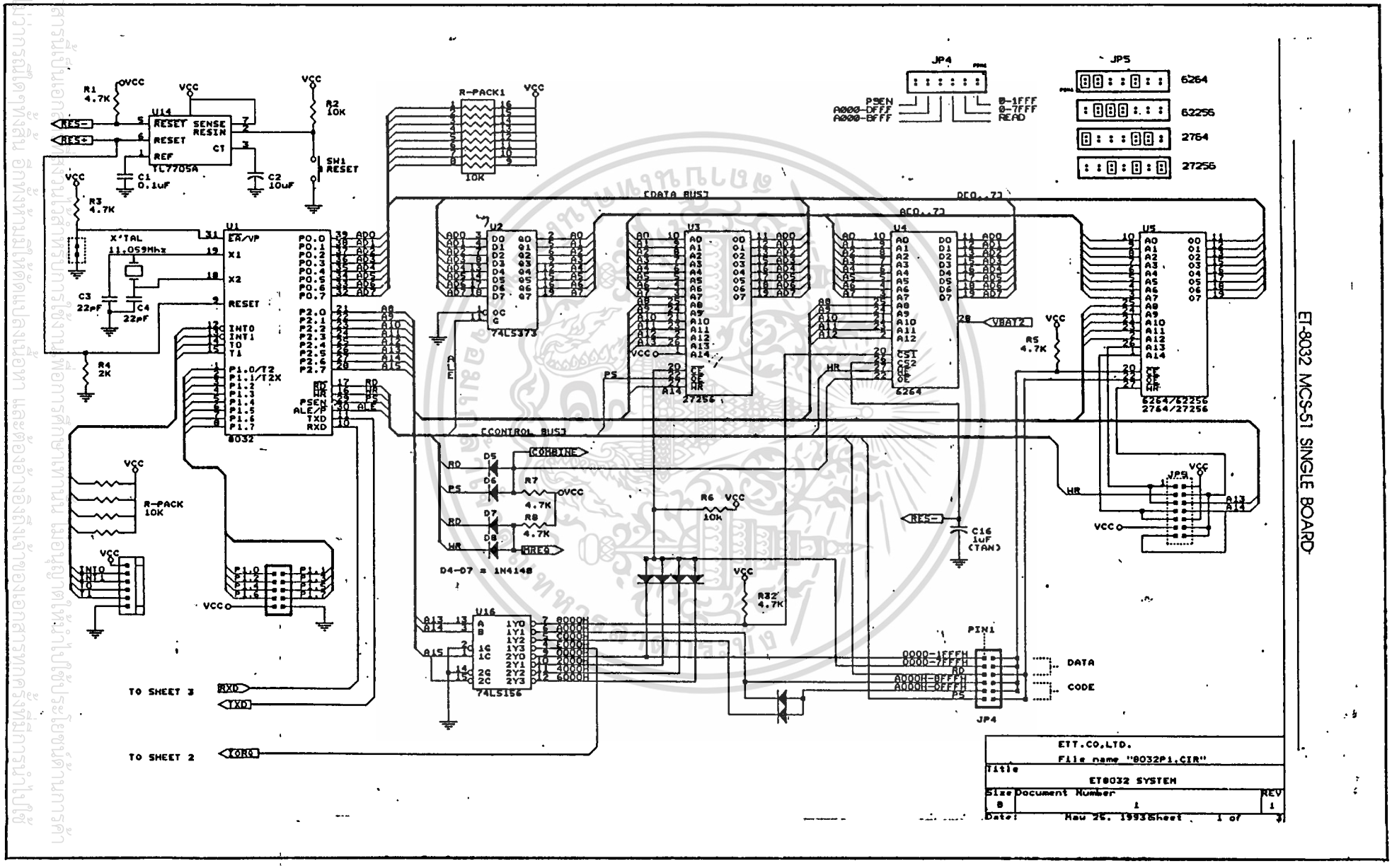
รูปที่ 4.8 แสดงบล็อกไดอะแกรมของตัวจำลองกระบวนการทางความร้อน

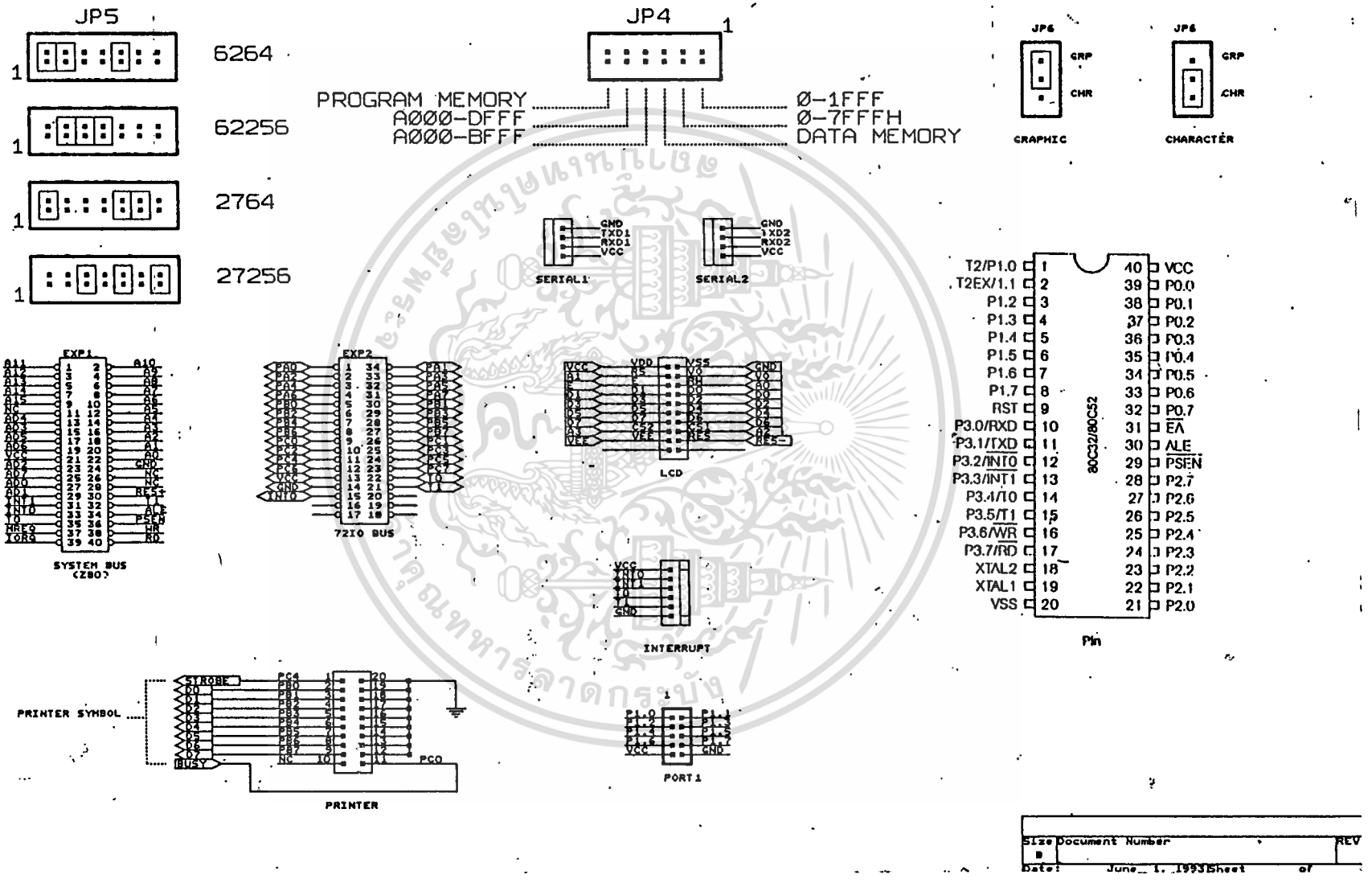
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

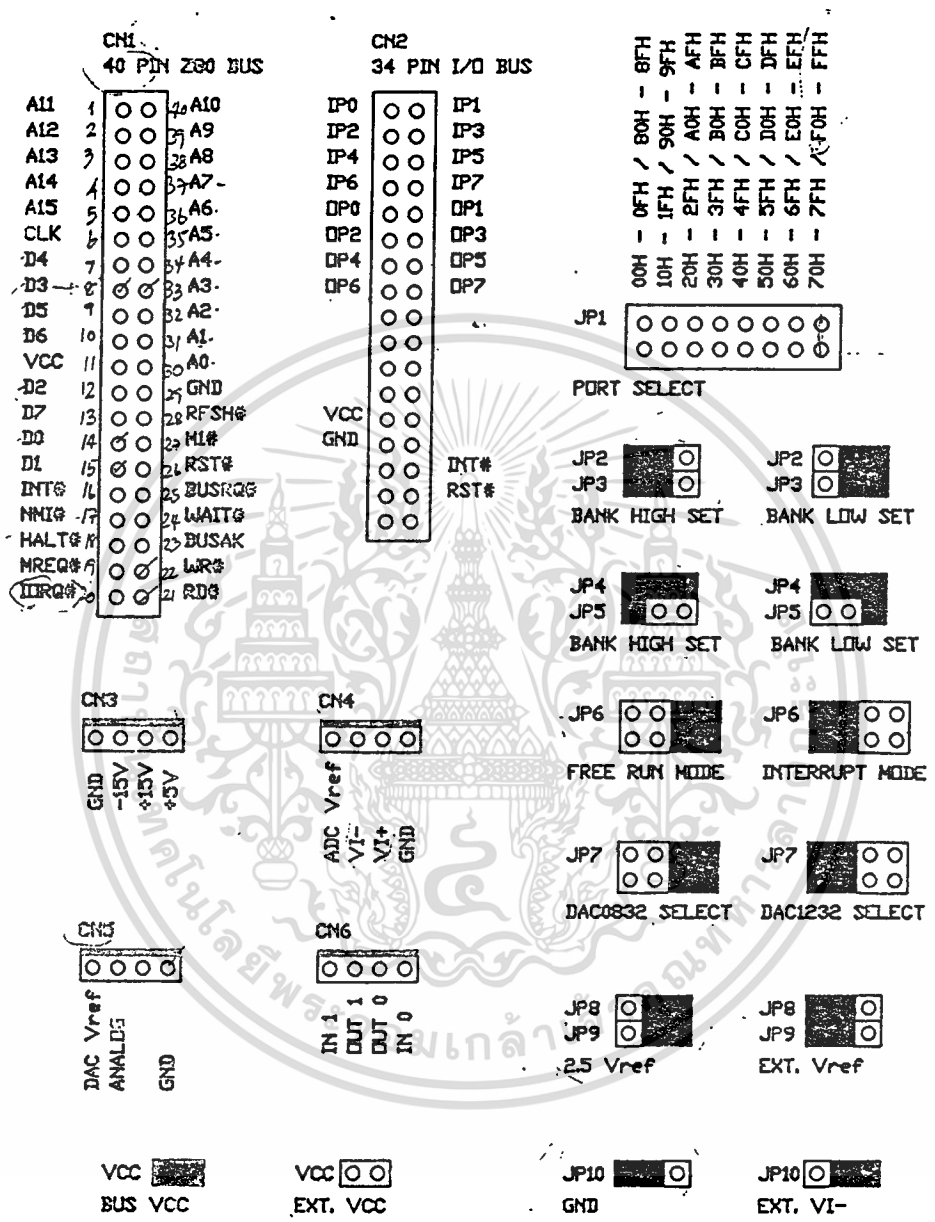


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


TXD
RXD
(SHEET 1)
C8 10uF
C9 10uF
C10 10uF
C11 10uF
U13
MAX232
VCC
GND
VCC RXD1 TXD1 GND
VCC RXD2 TXD2 GND
RXDP
TXDP
(SHEET 1)
U6
PA0 PA1 PA2 PA3 PA4 PA5 PA6 PA7
PB0 PB1 PB2 PB3 PB4 PB5 PB6 PB7
PC0 PC1 PC2 PC3 PC4 PC5 PC6 PC7
RD WR AO A1 RESET CS
8255
TO EXP2
EXP1
EXP2
I/O PORT
SYSTEM BUS
U6 CS
(SHEET 2)
E020H
CONTROL BUS
DATA BUS
PRINTER PORT (U6)
PRINTER SYMBOL
STROBE
BUSY
PC0
JP2
JP3
U12
RTC DS1207
RESET
IRQ
SQW
HOT
RTC
(SHEET 2)
Q1 BS170
R34 1K
R35 10
C15 470pF
LCD
EOCOH
(SHEET 2)
VR1 10K
CRP
CHR
JP8
R37 270
D26 LED
SW1
BREAK
INT0
SHEET 1
D24 1N4148
R-CHG 300
D25 1N4148
VBAT
3V BATTERY
J1 10VDC
D9 FR100
D21 1N4001
U17 LM7805
VI VO GND
C5 47uF
C6 470uF
C7 0.1uF
Z1 5.6V
R9 270
D10 LED
OVCC
ETT.CO,LTD.
File name "8032P3.CIR"
Title
I/O & CONNECTOR
Size B
Document Number 1
REV 1
Date: May 25, 1992 Sheet 3 of 3

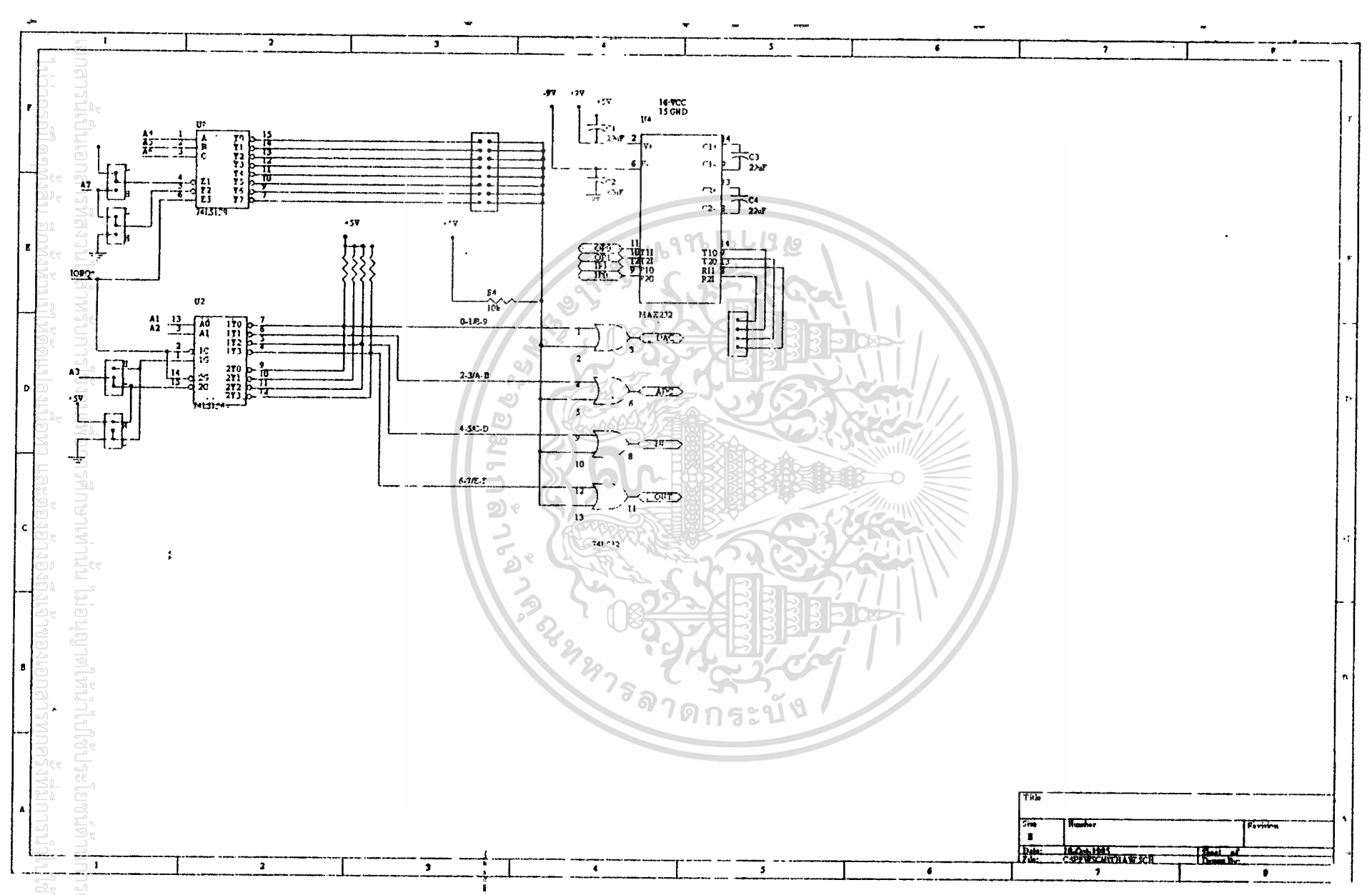

ET-8032 MCS-51 SINGLE BOARD
JP4
JP5
6264
62256
2764
27256
PSEN
A000-DFFF
A000-EFFF
8-1FFF
0-7FFF
READ
R-PACK1
10K
SH1
RESET
U14
RESET
SENSE
RESIN
REF
TL7705A
C1
0.1uF
C2
10uF
R1
4.7K
R2
10K
R3
4.7K
R4
2K
X'TAL
11.059Mhz
C3
22pF
C4
22pF
U1
8032
EA/VP
X1
X2
RESET
INT0
INT1
T0
T1
P1.0/T2
P1.1/T2X
PSEN
ALE/P
TXD
RXD
U2
74LS373
U3
27256
U4
6264
U5
6264/62256
2764/27256
[DATA BUS]
AD0..7
[CONTROL BUS]
COMBINE
D4-D7 = 1N4148
R5
4.7K
R6
10K
R7
4.7K
R8
4.7K
R32
4.7K
C16
1uF
(TAN)
VBAT2
U16
74LS156
0000H
2000H
4000H
6000H
8000H
A000H
C000H
E000H
0000-1FFFH
0000-7FFFH
A000H-BFFFH
A000H-DFFFH
DATA
CODE
PIN1
TO SHEET 3
TO SHEET 2
RXD
TXD
YORQ
ETT.CO,LTD.
File name "8032P1.CIR"
Title
ET8032 SYSTEM
Size
B
Document Number
1
REV
1
Date: Mau 26, 1993 Sheet 1 of 1


JP5
6264
62256
2764
27256
JP4
PROGRAM MEMORY
A000-DFFF
A000-BFFF
0-1FFF
0-7FFFH
DATA MEMORY
JP6
GRP
CHR
GRAPHIC
CHARACTER
SERIAL1
SERIAL2
GND
TXD1
RXD1
VCC
TXD2
RXD2
EXP1
SYSTEM BUS
(Z80)
EXP2
72IO BUS
LCD
INTERRUPT
PRINTER SYMBOL
STROBE
BUSY
PRINTER
PORT1
T2/P1.0
T2EX/1.1
P1.2
P1.3
P1.4
P1.5
P1.6
P1.7
RST
P3.0/RXD
P3.1/TXD
P3.2/INT0
P3.3/INT1
P3.4/T0
P3.5/T1
P3.6/WR
P3.7/RD
XTAL2
XTAL1
VSS
VCC
P0.0
P0.1
P0.2
P0.3
P0.4
P0.5
P0.6
P0.7
EA
ALE
PSEN
P2.7
P2.6
P2.5
P2.4
P2.3
P2.2
P2.1
P2.0
80C32/80C52
Pin
Size
Document Number
REV
Date:
June_ 1, 1993
Sheet
of

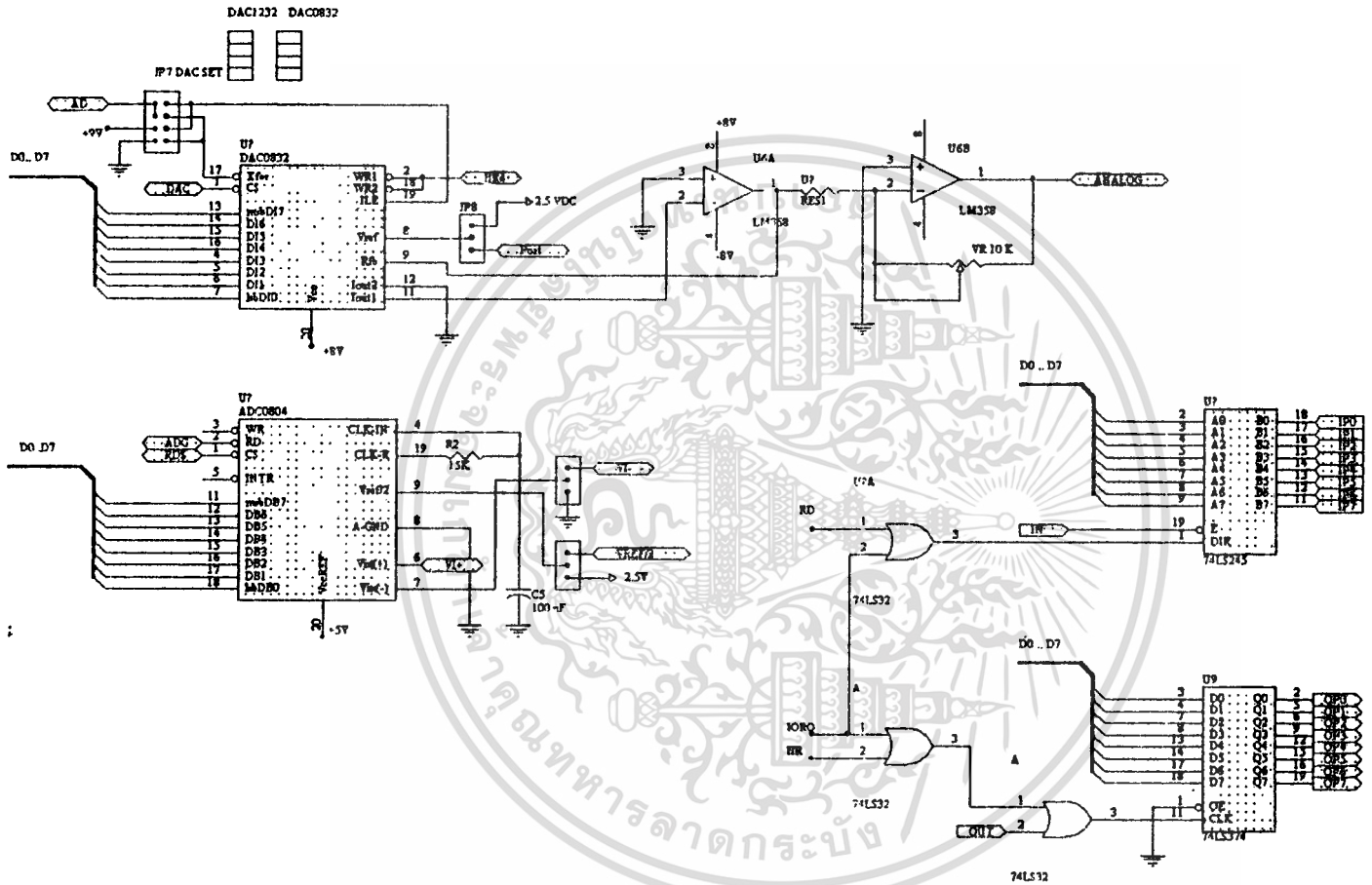


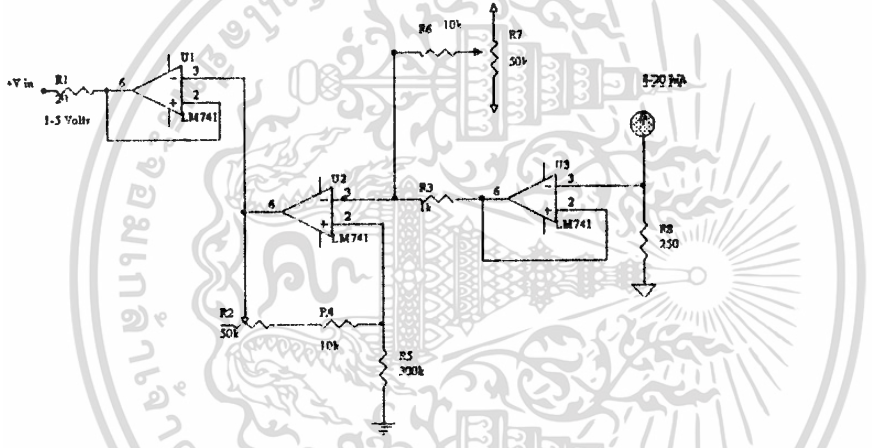
CONNECTOR REFERENCE & JUMPER SET-UP

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

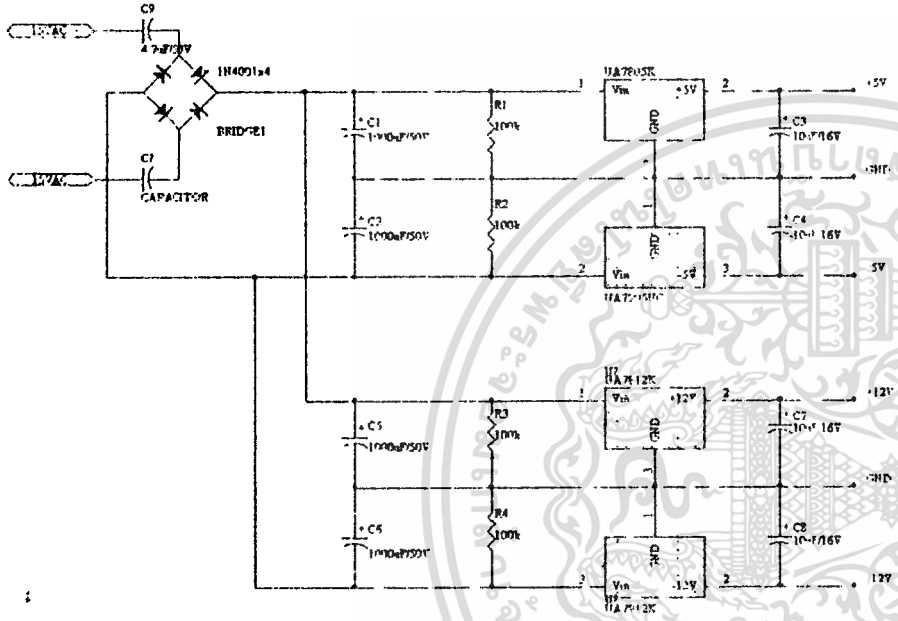


ลิขสิทธิ์ของมหาวิทยาลัยราชภัฏราชนครินทร์ ห้ามมิให้ทำซ้ำ ดัดแปลง เผยแพร่ หรือนำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ เว้นแต่ได้รับอนุญาตเป็นลายลักษณ์อักษรจากมหาวิทยาลัยราชภัฏราชนครินทร์ ขอสงวนสิทธิ์ในการดำเนินคดีตามกฎหมายอย่างเคร่งครัด

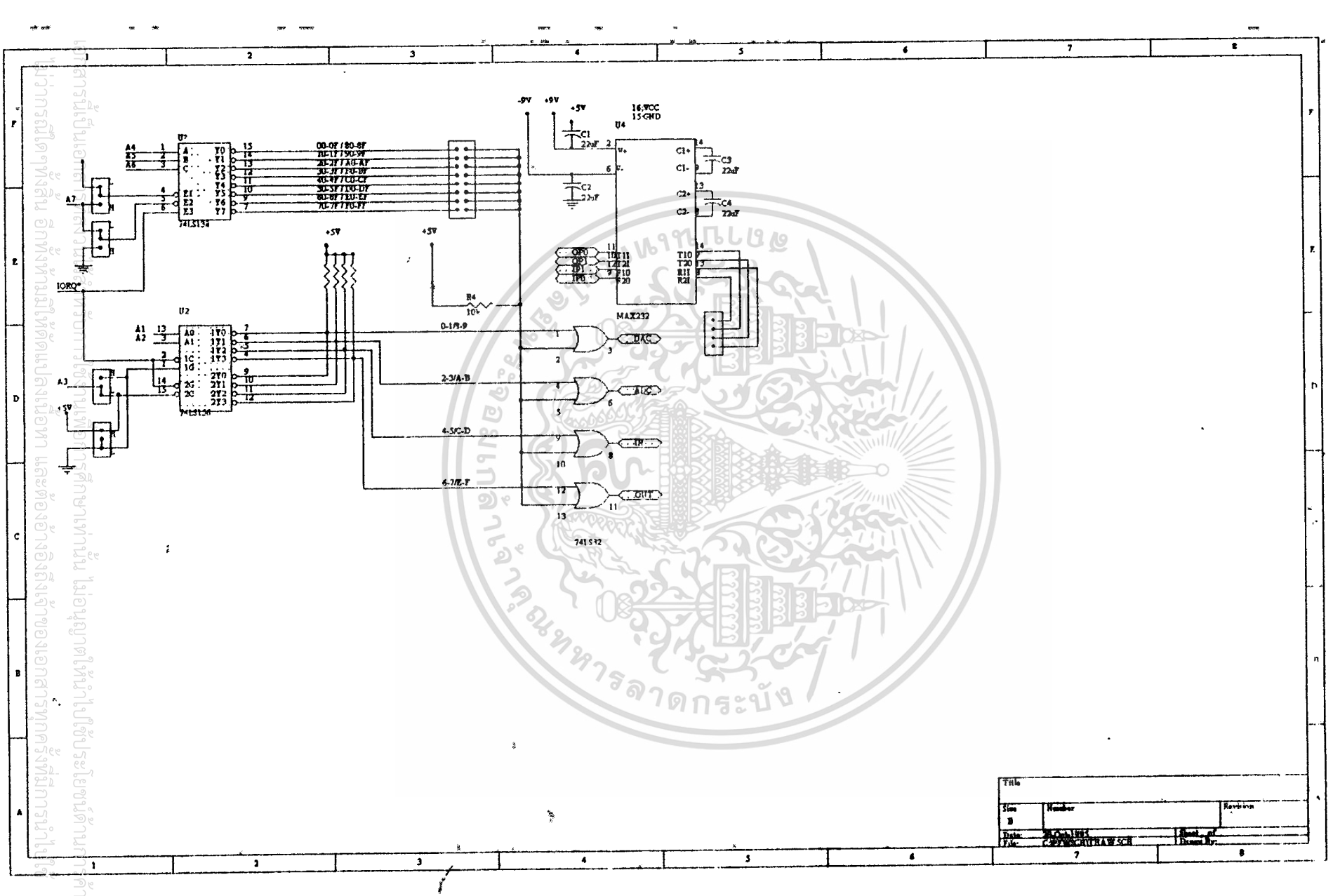

DAC1232 DAC0832
JP7 DAC SET
AD
DAC
U7
DAC0832
D0.. D7
WR1
WR2
XFER
ILE
CS
Vref
Rfb
Iout2
Iout1
JP8
2.5 VDC
Vref
+8V
-8V
U4A
LM358
U7
RES1
U6B
LM358
ANALOG
VR 10 K
U7
ADC0804
ADC
WR
RD
CS
INTR
CLK IN
CLK R
Vref/2
A GND
D GND
Vcc/REF
R2
15K
VC
C5
100 nF
2.5V
+5V
U7
74LS245
U7A
74LS32
IORQ
WR
U9
74LS374
OUT
74LS32



ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย



ภาควิชาวิศวกรรมไฟฟ้า กรมวิชาการทหารลาดกระบัง


U7
74LS154
U2
74LS139
U4
MAX232
16:VCC
15:GND
C1
22nF
C2
22nF
C3
22nF
C4
22nF
R4
10k
+5V
-9V
+9V
IORQ*
A7
A3
A4
A5
A6
A1
A2
00-0F / 80-8F
10-1F / 90-9F
20-2F / A0-AF
30-3F / B0-BF
40-4F / C0-CF
50-5F / D0-DF
60-6F / E0-EF
70-7F / F0-FF
0-1/8-9
2-3/A-B
4-5/C-D
6-7/E-F
DAC
ADC
741S42

เอกสารอ้างอิง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

1. เฮอร์เบิร์ต ชิลด์ การประยุกต์ใช้งานภาษาซี. บริษัท ซีเอ็ดยูเคชั่น จำกัด

2. ชูชัย ธนสารตั้งเจริญ และทินกร ดุ๊ก การสื่อสารข้อมูล. สถาบันเทคโนโลยีพระจอมเกล้า เจ้าคุณทหารลาดกระบัง

3. ธันวา ศรีประโมง การเขียนโปรแกรมภาษาซี สำหรับวิศวกรรม. มหาวิทยาลัยเทคโนโลยีมหานคร พิมพ์ครั้งที่ 3 , 2537

4. เข้าใจ/สร้าง/เล่น ไมโครโปรเซสเซอร์ เล่ม 1. บริษัท ซีเอ็ดยูเคชั่น จำกัด

5. สุนทร วิทูสุรพจน์ การใช้งานไมโครคอนโทรลเลอร์ ตระกูล 8051. บริษัท ซีเอ็ดยูเคชั่น จำกัด , 2537

6. สุนทร วิทูสุรพจน์ การโปรแกรมภาษาแอสเซมบลี ของไมโครคอนโทรลเลอร์ ตระกูล 8051. บริษัท ซีเอ็ดยูเคชั่น จำกัด , 2537

7. ET-8032 V2.0 MCS-51 SINGLE BOARD MICRO CONTROLLER USE ' S MANUAL. ETT CO.,LTD.

8. JONH WILEY $ SONS PRINCIPLES AND PRACTICE OF AUTOMATIC PROCESS CONTROL. , 1995

9. Ayala , K.J. The 8051 Microcontroller Architectecture , Programing and Application. West Publishing Company , 1991

10. Benjamin C. Kuo Automatic Control System. Third Edition Prentice - Hall , India , 1976

11. Katsuhiko Ogata Modern Control Engineering. Prentice - Hall , India , 1978

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้