ระบบควบคุมการเข้าออก ด้วยบัตรแม่เหล็ก และ คิดเงินเดือนอัตโนมัติ

Magnetic Card Access Control System and Automatic Salary Calculator

โดย

นาย บุญเลิศ รุจน์เรชา

นาย บุญเรือง สกุลเดชธนา

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาอุตสาหกรรมศาสตร์บัณฑิต
สาขาเทคโนโลยีอิเล็กทรอนิคส์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2537

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญานิพนธ์ ระบบควบคุมการเข้าออกด้วยบัตรแม่เหล็ก และคิดเงินเดือนอัตโนมัติ

Magnetic Card Access Control System and Automatic Salary Calculator

ชื่อนักศึกษา นาย บุญเลิศ รุจน์เรขา

นาย บุญเรือง สกุลเดชธนา

อาจารย์ที่ปรึกษา อาจารย์ชวลิต เบญจางคประเสริฐ

ภาควิชา เทคนิคอุตสาหกรรม

ปีการศึกษา 2537

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง อนุมัติให้นับปริญญานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรอุตสาหกรรมศาสตร์บัณฑิต

คณะกรรมการสอบปริญญานิพนธ์

..... ประธานกรรมการ

( )

..... กรรมการ

( )

..... กรรมการ

( )

..... กรรมการ

( )

..... กรรมการ

( )

ลิขสิทธิ์ของคณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้า เจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบควบคุมการเข้าออกด้วยบัตรแม่เหล็ก และคิดเงินเดือนอัตโนมัติ

Mgnetic Card Access Control System and Automatic Salary Calculator

โดย

1. นาย บุญเลิศ รุจน์เรขา
2. นาย บุญเรือง สกุลเดชธนา

อาจารย์ที่ปรึกษา อาจารย์ชวลิต เบญจางคประเสริฐ

ปีการศึกษา 2537

บทคัดย่อ

โครงงานที่ทำการค้นคว้า ออกแบบ และพัฒนานี้ เป็นการใช้บัตรแม่เหล็กในการ ผ่านเข้าออก และทำการบันทึกเวลาผ่านเข้าออกของพนักงาน ในที่ทำงาน โดยบัตรแม่เหล็กที่ใช้นี้ จะเป็นบัตรเอทีเอ็มที่ใช้ในธนาคารต่าง ๆ ซึ่งจะเป็นมาตรฐานเดียวกันทั้งหมด โดยในโครงสร้างของเครื่องนี้จะมีส่วนของ Center ส่วนของ Terminal และ Computer โดย Center จะเป็นศูนย์กลางของระบบ จะมีการติดต่อข้อมูลกับ Terminal และ Computer โดยในการติดต่อกับ Terminal จะเป็นระบบที่ใช้ทำการตรวจสอบการเข้าออก และการบันทึกเวลาทำงานของพนักงาน โดยจะใช้เป็นระบบเครือข่าย โดยมีตัว Center เป็นศูนย์กลางของระบบซึ่งจะทำการควบคุมการรับส่งข้อมูลภายในระบบทั้งหมด โดยจะใช้มาตรฐานการสื่อสาร RS-485 ในการรับส่งข้อมูล ส่วนในการติดต่อกับ Computer จะเป็นการนำข้อมูลของเวลาเข้าออกของพนักงานส่งไปให้ Computer เพื่อทำการเก็บข้อมูลไว้ทั้งหมด ซึ่งจะรับส่งข้อมูลผ่านทางมาตรฐานการสื่อสาร RS-232

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Magnetic Card Access Control System and Automatic Salary Cauculator

by

1. Mr. Boonlert Rutrekha
2. Mr. Boonruang Sakuldechthana

Advisor Mr. Chauvalit Benjangkaprasert

Abstract

This project is use the magnetic card to pass in or pass out in office or factory and record work time of employee. The magnetic card use the ATM card of the bank which is same standard.

This system comprise Center and Terminal which use the RS-485 network system. The Center will be control all data that communicate in system. Center will poll to Terminal to receive data from Terminal.

It's use cassette head for read data in this magnetic card. Then data will be amplify and decode card data in the micro processor . This data will be transmitting to the center. The center is receive worktime , day and data for only day. Finally the center will be transmitting worktime , day and date to the computer for calculated worktime and salary of employee.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้ได้สำเร็จลุล่วงไปได้ด้วยด้วยดีด้วยความช่วยเหลือ ของ อาจารย์ชวลิต เบญจางคประเสริฐ อาจารย์ที่ปรึกษาวิทยานิพนธ์ ซึ่งได้ให้คำแนะนำและข้อเสนอแนะต่าง ๆ ในการสร้างเครื่องนี้ และยังขอขอบคุณ อาจารย์พลผดุง ผดุงกุล ซึ่งท่านเป็นอาจารย์ประจำภาควิชาวิศกรรมอิเล็กโทรนิคส์ ซึ่งท่านได้ให้ข้อมูล ตลอดจนคำแนะนำในเรื่องของบัตรแม่เหล็กของธนาคาร

ท้ายนี้ ผู้วิจัยใคร่ขอกราบขอบพระคุณ บิดา-มารดา ซึ่งสนับสนุนในด้านการเงิน และให้กำลังใจแก่ ผู้วิจัยตลอดมา จนสามารถทำการวิจัย และสร้างเครื่องนี้สำเร็จลงได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

ในปัจจุบันนี้ระบบการตอกบัตรเข้าทำงานมักจะพบเห็นได้ตามบริษัททั่ว ๆ ไป ที่ต้อการจะบันทึกเวลาเข้าออกทำงานของพนักงานไว้ และนำค่าเวลาที่บันทึกได้นี้มาคิดเป็นเงินเดือนและจ่ายให้พนักงานเป็นคน ๆ ไป ซึ่งระบบการบันทึกเวลานี้ถ้าเป็นวิธีเก่าหน่อย ก็คือจะเป็นการพิมพ์ว่าเวลาลงไปที่บัตร แต่ถ้าเป็นระบบใหม่หน่อย ก็เช่นวิธีการใช้แถบรหัสบาร์โค้ด หรือ บัตรแม่เหล็กในการตรวจสอบเวลาที่เข้าออกทำงาน ซึ่งทั้งสองแบบหลังนี้ ค่อนข้างจะมีความสะดวกและความน่าเชื่อถือดีกว่า และยังสามารถตรวจสอบไม่ให้บุคคลภายนอกที่ไม่มีบัตร เข้ามาในสำนักงานหรือที่ทำงานนั้น ๆ ได้อีกด้วย

ในโครงงานนี้จะใช้บัตรแม่เหล็ก ซึ่งเป็นมาตรฐานเดียวกันกับบัตร เอทีเอ็ม ของธนาคารในการตรวจสอบเวลาเข้าออก ของพนักงาน เหตุที่ใช้บัตรแม่เหล็กมาตรฐานเดียวกันกับบัตร เอทีเอ็ม เนื่องจากต้องการให้สามารถพัฒนาระบบต่อไปให้เชื่อมต่อข้อมูลกับธนาคาร เพื่อให้สามารถโอนเงินเข้าบัญชีผ่านทางธนาคารได้อย่างสะดวก

ลักษณะการใช้งานของเครื่องนี้ จะใช้บัตรแม่เหล็กผ่านเข้าออกทำงาน ซึ่งจะมีการบันทึกเวลาเข้าออกไว้ เมื่อสิ้นสุดการทำงานในวันหนึ่ง ก็จะทำการส่งค่าเวลาเข้าออกไปให้ตัวเก็บข้อมูล ซึ่งก็คือคอมพิวเตอร์ เมื่อสิ้นสุดการทำงานในแต่ละเดือนหรือวันที่จะมีการจ่ายเงินเดือนให้พนักงาน ในส่วนของคอมพิวเตอร์ก็จะทำการคำนวณเงินเดือนที่พนักงานจะได้รับออกมา

โครงสร้างของระบบนี้จะประกอบไปด้วยอุปกรณ์ต่าง ๆ ดังแสดงในรูปที่ 1.1

จากโครงสร้างของระบบจะประกอบไปด้วย

1. คอมพิวเตอร์ จะมีหน้าที่ในการเก็บข้อมูลของค่าเวลาที่พนักงานเข้าออกทำงานทั้งหมด โดยรับข้อมูลมาจากส่วนของ Center โดยจะรับข้อมูลทุก ๆ วันที่มีการทำงาน จากนั้นเมื่อถึงวันที่จะต้องมีการจ่ายเงินเดือน คอมพิวเตอร์ก็จะทำการคำนวณเงินเดือนของพนักงานแต่ละคนออกมา นอกจากนี้แล้วยังสามารถดูค่าเวลาเข้าออก ของพนักงานและพิมพ์ออกทางเครื่องพิมพ์ได้ด้วย

2. Center ทำหน้าที่เป็นศูนย์กลางของระบบ ที่จะตรวจสอบการผ่านเข้าออก และการบันทึกเวลาเข้าออกของพนักงาน ซึ่ง Center จะเก็บข้อมูลของบัตรแม่เหล็กที่จะเข้าออกเอาไว้ทั้งหมดและจะทำการตรวจสอบว่าบัตรใบใด สามารถเข้าออกได้ หรือไม่ได้ และยังตรวจสอบเวลาที่จะมีการบันทึกเวลาเข้าออกด้วย โดยบัตรที่จะทำการเข้าออก จะติดต่อมาทาง Terminal และ Center จะส่งผลที่ได้กลับไปให้ Terminal ให้ทำงานตามสั่ง

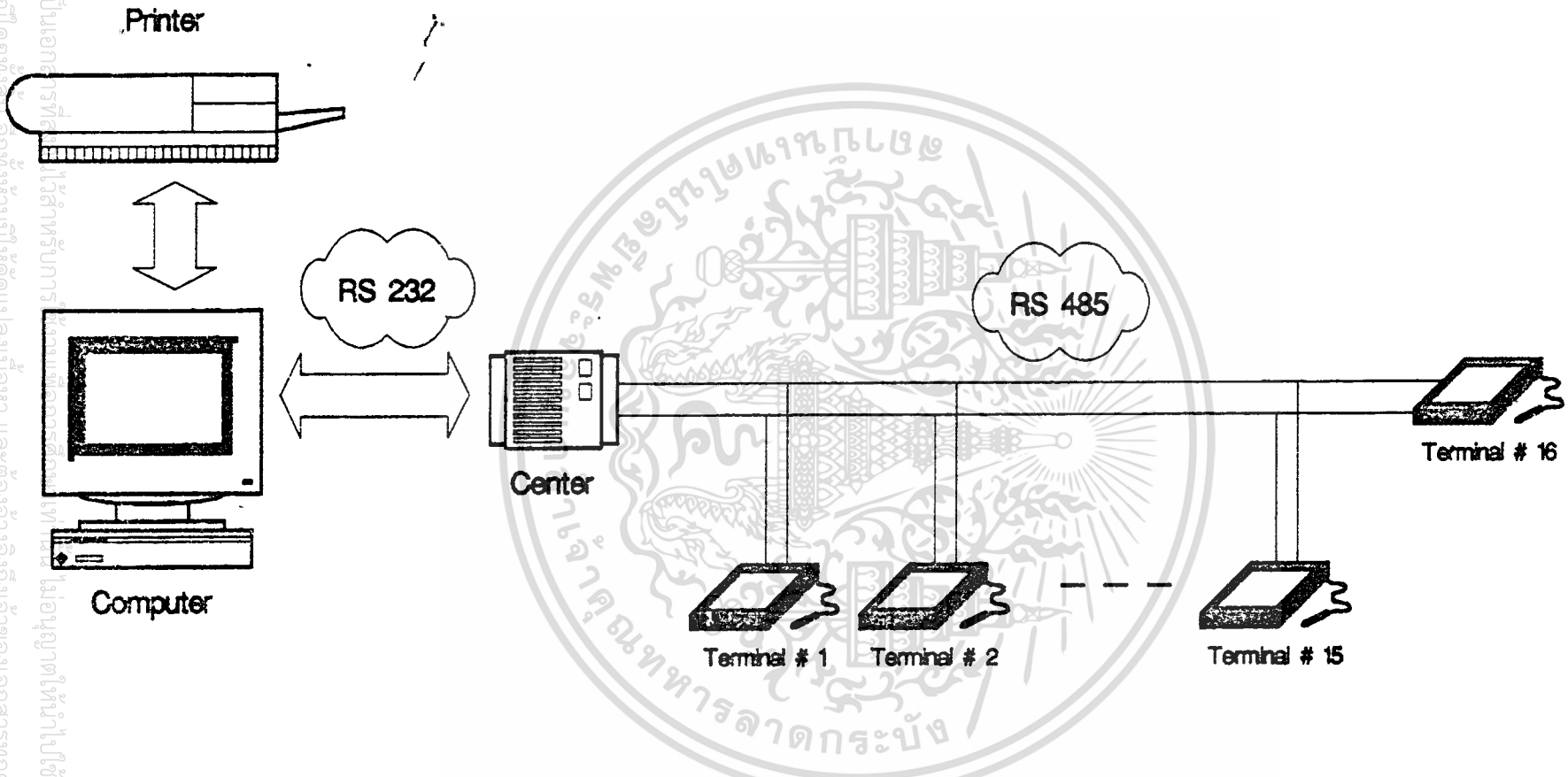
3. Terminal มีหน้าที่ในการควบคุมการเปิดประตูที่จะให้พนักงาน หรือบุคคลต่าง ๆ ที่จะผ่านเข้าออกในบริเวณนั้น ๆ โดย Terminal นี้จะมีจำนวนได้หลายตัวซึ่งจะติดตั้งอยู่ที่บริเวณที่จะมีการเข้าออกได้หลายที่ โดยในระบบนี้จะสามารถต่อ Terminal ได้สูงสุด 16 ตัว ตามรูปที่แสดงโครงสร้างของ

เอกสารนี้ถูกใช้ได้หลายที่ โดยในระบบนี้จะสามารถต่อ Terminal ได้สูงสุด 16 ตัว ตามรูปที่แสดงโครงสร้างของ
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบ โดยแต่ละตัวจะตรวจสอบการรูดบัตรเพื่อจะเข้าออก และส่งข้อมูลของบัตรนั้นไปให้ Center เพื่อให้ Center ทำการตรวจสอบ และส่งผลที่ได้กลับมา

ในการติดต่อข้อมูลระหว่าง คอมพิวเตอร์ กับ Center จะใช้มาตรฐานการสื่อสารข้อมูลแบบ RS-232 ซึ่งเป็นมาตรฐานที่ใช้ในการรับส่งข้อมูลแบบ Full Duplex และการติดต่อระหว่าง Center กับ Terminal จะใช้มาตรฐานการสื่อสารแบบ RS-485 ซึ่งเป็นการรับส่งข้อมูลแบบ Half Duplex ซึ่งมาตรฐานทั้ง 2 ชนิดนี้จะอธิบายอย่างละเอียดในบทที่ 3 เรื่องการติดต่อสื่อสารข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ๒ แสดงโครงสร้างของระบบ

บทที่ 2

บัตรแม่เหล็ก

บัตรแม่เหล็กที่ใช้ในงานนี้ จะใช้บัตรที่เป็นมาตรฐานเดียวกันกับมาตรฐานบัตรเอทีเอ็มของธนาคาร ซึ่งมีคุณลักษณะของบัตรเป็นไปตามมาตรฐาน ISO 7811/2 และ /4 (1985) เหตุที่ใช้บัตรที่เป็นมาตรฐานเดียวกันกับบัตรเอทีเอ็มของธนาคารนี้ เนื่องจากต้องการให้ในการพัฒนาระบบต่อไป สามารถใช้บัตรนี้ร่วมกันกับธนาคารโดยตรง โดยจะสามารถโอนเงินเข้าไปในบัญชีที่พนักงานเปิดบัญชีอยู่ในบัตรเอทีเอ็มที่นำมาใช้ เพื่อความสะดวกในการจ่ายเงินให้พนักงาน

มาตรฐานของบัตร

ในมาตรฐานของบัตรนี้ จะแบ่งการเก็บข้อมูลของบัตรไว้เป็น 3 แทรก โดยจะมีรายละเอียดแต่ละแทรกคือ

- แทรกที่ 1 เป็นแทรกที่ใช้สำหรับการอ่าน ซึ่งเก็บข้อมูลที่ประกอบด้วย ตัวอักษรและตัวเลข
- แทรกที่ 2 เป็นแทรกที่ใช้สำหรับการอ่าน ซึ่งเก็บข้อมูลที่ประกอบด้วยตัวเลขอย่างเดียว
- แทรกที่ 3 เป็นแทรกที่ใช้สำหรับการอ่านและเขียน ซึ่งเก็บข้อมูลที่ประกอบด้วยตัวเลข

ในการใช้งานของเครื่องนี้ จะใช้แทรกที่ 2 เป็นข้อมูลประจำตัวของพนักงานที่จะเข้าออก เนื่องจาก เป็นแทรค ที่ใช้เก็บข้อมูลของธนาคาร และเลขที่บัญชี ดังนั้นจึงเป็นแทรค ที่แต่ละธนาคารจะสามารถใช้ร่วมกันได้ ส่วนในแทรกที่ 1 และ 3 ที่แต่ละธนาคารจะใช้เป็นที่เก็บข้อมูลต่าง ๆ ที่ไม่เหมือนกันทั้งหมด ดังนั้นจึงใช้ทั้ง 2 แทรคนี้ไม่ได้ โดยลักษณะของข้อมูลของแทรกที่ 2 จะแสดงดังรูปที่ 2.1

Sync	Start (B)	Data (0-9)	Sep (D)	Data (0-9)	Stop (F)	LRC	Sync

รูปที่ 2.1 แสดงรูปแบบข้อมูลที่อยู่ในแทรกที่ 2 ของบัตรเอทีเอ็ม

จากรูปที่ 2.1 ข้อมูลที่อยู่ในแทรกที่ 2 ของบัตรจะประกอบไปด้วย

- Sync เป็นส่วนเริ่มต้นและส่วนท้ายของข้อมูล โดย Sync จะมีข้อมูลเป็นบิท "0"ทั้งหมดจำนวนหลายบิท เพื่อใช้เป็นตัวกำหนดความถี่ของข้อมูลบิท "0" และให้การอ่านข้อมูลเริ่มที่ตำแหน่งที่ถูกต้อง

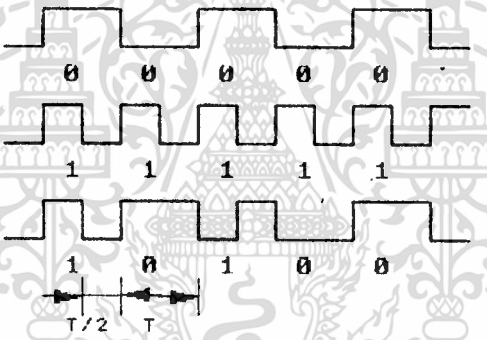
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Start เป็นไบต์เริ่มต้นของข้อมูล ซึ่งมีค่าเป็น 0BH (ค่าเลขฐาน 16)
- Data เป็นส่วนของข้อมูล โดยมีค่าเป็นตัวเลข 0-9
- Sep เป็นส่วนที่ใช้คั่นข้อมูลในส่วนของ Data โดยมีค่าเป็น 0DH
- Stop เป็นไบต์สิ้นสุดข้อมูล ซึ่งมีค่าเป็น 0FH
- LRC เป็นไบต์ตรวจสอบการผิดพลาดของข้อมูลทั้งหมด

ข้อมูลจากไบต์ Start ไปจนถึง CRC จะมีจำนวนทั้งหมด 40 ไบต์

<u>ลักษณะสัญญาณที่บันทึกในบัตร</u>

สัญญาณที่บันทึกในบัตร จะเป็นลักษณะของสัญญาณดิจิตอล โดยลักษณะของสัญญาณที่บันทึกอยู่ในบัตรแม่เหล็กแสดงได้ดังรูปที่ 2.2



รูปที่ 2.2 ลักษณะของสัญญาณที่บันทึกอยู่ในบัตรแม่เหล็ก

จากรูปที่ 2.2 ลักษณะของสัญญาณ "0" จะมีค่าความถี่ต่ำกว่าสัญญาณ "1" อยู่ 2 เท่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อมูลในแทรกที่ 2

ข้อมูลที่บรรจุอยู่ในแทรกที่ 2 แสดงได้ดังตารางที่ 2.1

Bits					ROW	Char
P	b4	b3	b2	b1		
1	Ø	Ø	Ø	Ø	0	Ø
Ø	Ø	Ø	Ø	1	1	1
Ø	Ø	Ø	1	Ø	2	2
1	Ø	Ø	1	1	3	3
Ø	Ø	1	Ø	Ø	4	4
1	Ø	1	Ø	1	5	5
1	Ø	1	1	Ø	6	6
Ø	Ø	1	1	1	7	7
Ø	1	Ø	Ø	Ø	8	8
1	1	Ø	Ø	1	9	9
1	1	Ø	1	Ø	10	A
Ø	1	Ø	1	1	11	B
1	1	1	Ø	Ø	12	C
Ø	1	1	Ø	1	13	D
Ø	1	1	1	Ø	14	E
1	1	1	1	1	15	F

ตารางที่ 2.1 ข้อมูลในแทรกที่ 2

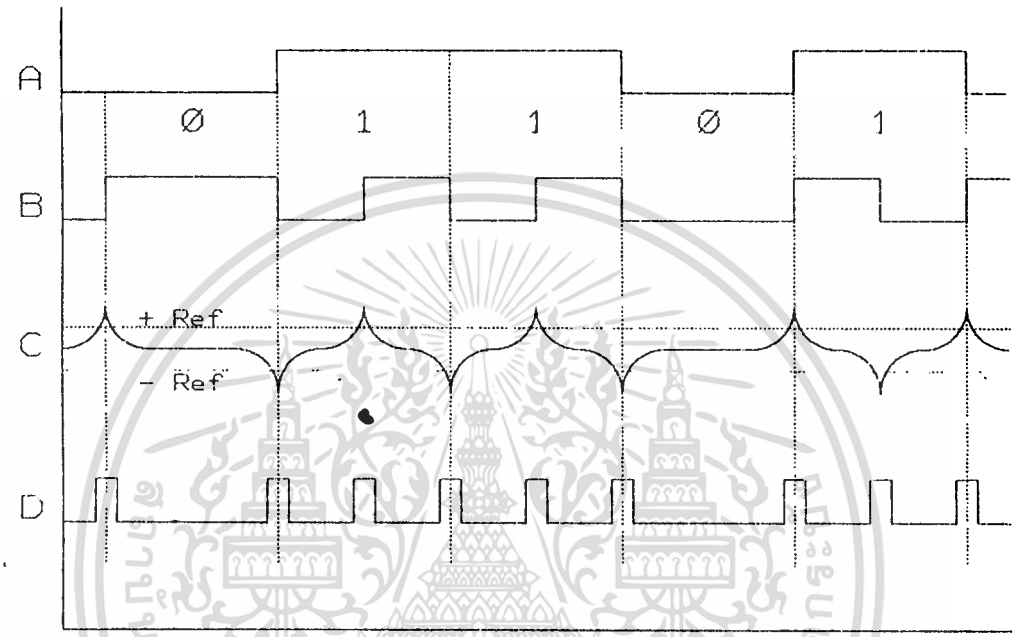
จากตารางที่ 2.1 ข้อมูล 1 character จะได้จากบิทข้อมูล 4 บิท และ parity 1 บิท รวมเป็น 5 บิท โดย Parity จะเป็นแบบ odd Parity และข้อมูล 4 บิทจะมีค่าตรงกับเลขฐาน 16 โดยค่า 0-9 จะเป็นข้อมูล และค่า A-F จะเป็น Control Charactors โดยมีรายละเอียดคือ

- B เป็นตัวแสดงจุดเริ่มต้นของข้อมูล (Start)
- D เป็นตัวแยกข้อมูล (sep)
- F เป็นตัวแสดงจุดสิ้นสุดของข้อมูล (Stop)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ลักษณะสัญญาณที่ใช้ในการถอดรหัส

สัญญาณต่าง ๆ ในการถอดรหัสของบัตร แสดงได้ดังรูปที่ 2.3



รูปที่ 2.3 สัญญาณต่าง ๆ ในการถอดรหัสของบัตร

จากรูปที่ 2.3 จะได้สัญญาณต่าง ๆ ดังต่อไปนี้

- รูป A เป็นข้อมูลจริงที่จะต้องทำการ Decode ออกมา
- รูป B เป็นสัญญาณมาตรฐานของบัตรที่บันทึกอยู่ในบัตร
- รูป C เป็นสัญญาณที่อ่านได้จากบัตรแม่เหล็ก
- รูป D เป็นสัญญาณที่ได้จากการนำสัญญาณที่อ่านได้จากบัตรแม่เหล็ก (รูป 4ค) มาผ่านวงจรเปรียบเทียบแรงดัน เพื่อนำส่วนของสัญญาณที่สูงกว่าค่าแรงดันเปรียบเทียบทั้งในทางด้านบวกและสัญญาณส่วนที่ต่ำกว่าค่าแรงดันเปรียบเทียบทางด้านลบ มาสร้างเป็นสัญญาณพัลส์ โดยในการ Decode ข้อมูลในบัตร จะใช้สัญญาณนี้ไปทำการ Decode

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การถอดรหัสข้อมูลในบัตร

ในการถอดรหัสข้อมูลในบัตรจะแบ่งขั้นตอนออกเป็น 3 ขั้นตอนคือ

1. การอ่านบัตร

ในการอ่านบัตรแม่เหล็กนี้ จะใช้หัวอ่านเป็นหัวเทปที่ใช้ใน Tape Cassette ทั่วไป โดยจะใช้เป็นหัวแบบโมโน เนื่องจากจะมีความกว้างของแทรกมากกว่าหัวเทปแบบสเตอริโอ โดยในการอ่านบัตรนี้ จะต้องสร้างตัวเครื่องที่ใช้ในการอ่านบัตรขึ้นเอง และจะสร้างตัวอ่านออกเป็น 2 แบบคือ แบบใช้มือรูด และแบบใช้เครื่องอ่านอัตโนมัติ ซึ่งแสดงได้ดังรูปที่ 2.4 และมีรายละเอียดดังนี้

1.1 แบบใช้มือรูด

เครื่องอ่านบัตรแบบใช้มือรูดจะทำเป็นช่องสำหรับใส่บัตรเพื่อทำการรูด โดยจะมีความลึกประมาณครึ่งหนึ่งของความกว้างของบัตร และมีความยาวพอที่จะอ่านบัตรได้ตลอดความยาวของแถบแม่เหล็ก

1.2 แบบใช้เครื่องอ่านอัตโนมัติ

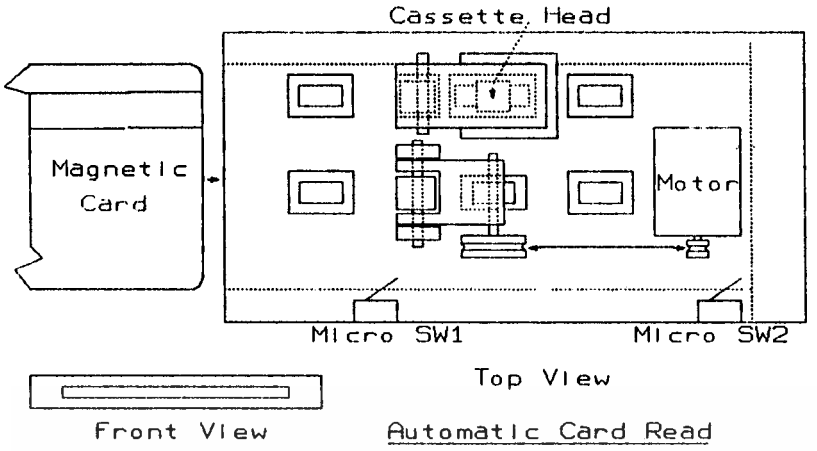
เครื่องอ่านบัตรแบบอัตโนมัติจะใช้มอเตอร์ เป็นตัวเลื่อนบัตรผ่านเข้าไปเพื่อทำการอ่าน โดยในส่วนของหัวเทปจะอยู่กับที่ วงจรที่ใช้ในการควบคุมการเลื่อนบัตรเข้าออก แสดงได้ดังรูปที่ 2.5

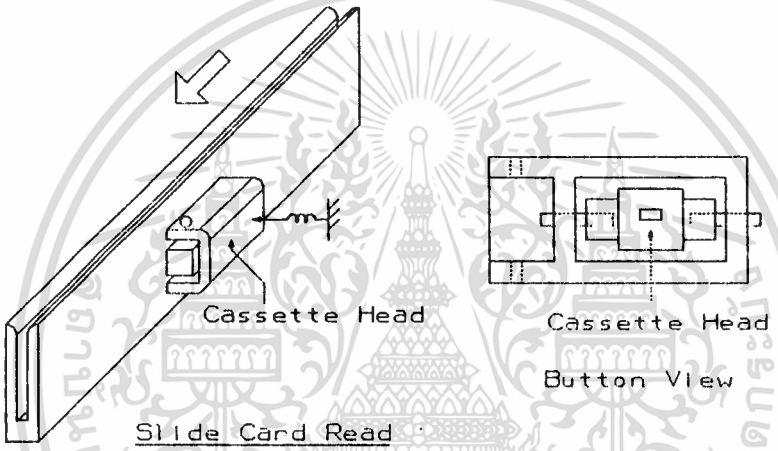
จากรูปที่ 2.5 ซึ่งวงจรควบคุมการเลื่อนบัตรเข้าออก จะใช้ Transistor เป็นตัวนำกระแสให้ Motor ทำงาน โดยจะควบคุมทิศทางการหมุนของ Motor ถ้าจะทำการเลื่อนบัตรเข้าไปอ่านจะต้องทำการเซ็ท IC # 4027 ให้ทำงาน ซึ่งจะได้เอาท์พุทมาทำการเซ็ทจาก Port out ของ CPU ส่วนในการเลื่อนบัตรออกจะต้องทำการทริก IC เบอร์ 4528 ซึ่งเป็นวงจรตั้งเวลา ให้ทำงาน โดยจะมีไมโครสวิทช์จำนวน 2 ตัวในการตรวจสอบการทำงาน โดยตัวหนึ่งจะทำงานเมื่อมีบัตรใส่เข้ามาหรือมีบัตรค้างอยู่ในเครื่องอ่าน ส่วนอีกตัวหนึ่งจะตรวจสอบจุดสิ้นสุดของบัตรในการเลื่อนบัตรเข้ามาถึงข้างในสุด เพื่อจะทำให้ Motor หยุดหมุน และเป็นจุดสิ้นสุดการอ่านบัตร

ในส่วนของชุดหัวเทปที่ใช้ในการอ่านจะมีโครงสร้างแสดงได้ดังรูปที่ 2.4

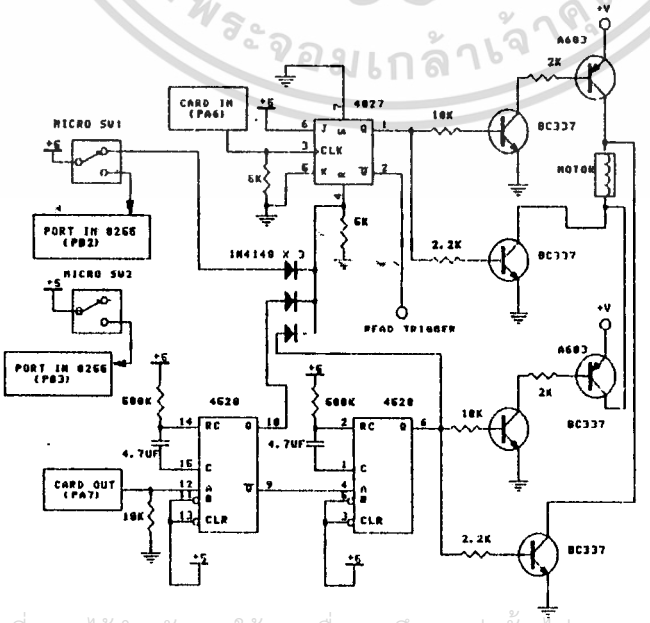
จากรูปที่ 2.4 ลักษณะของชุดหัวเทปที่ใช้อ่านบัตร จะต้องสามารถทำให้หัวเทปแนบสนิทกับบัตรในเวลาอ่านได้ตลอดช่วงของการอ่าน ดังนั้นชุดหัวเทปจะต้องมีลักษณะเป็นสปริงดันชุดหัวเทปให้กดลงที่บัตรดังรูป และหัวเทปจะต้องมีการหมุนได้เล็กน้อยเพื่อให้หัวเทปแนบสนิทกับบัตรตลอดช่วงเวลาการอ่าน ซึ่งในส่วนนี้จะต้องทำการออกแบบและสร้างขึ้นเอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้





รูปที่ 2.4 เครื่องอ่านบัตรแบบอัตโนมัติ และแบบใช้มือรูด



รูปที่ 2.5 แสดงวงจรที่ใช้ควบคุมเครื่องอ่านบัตรแบบอัตโนมัติ

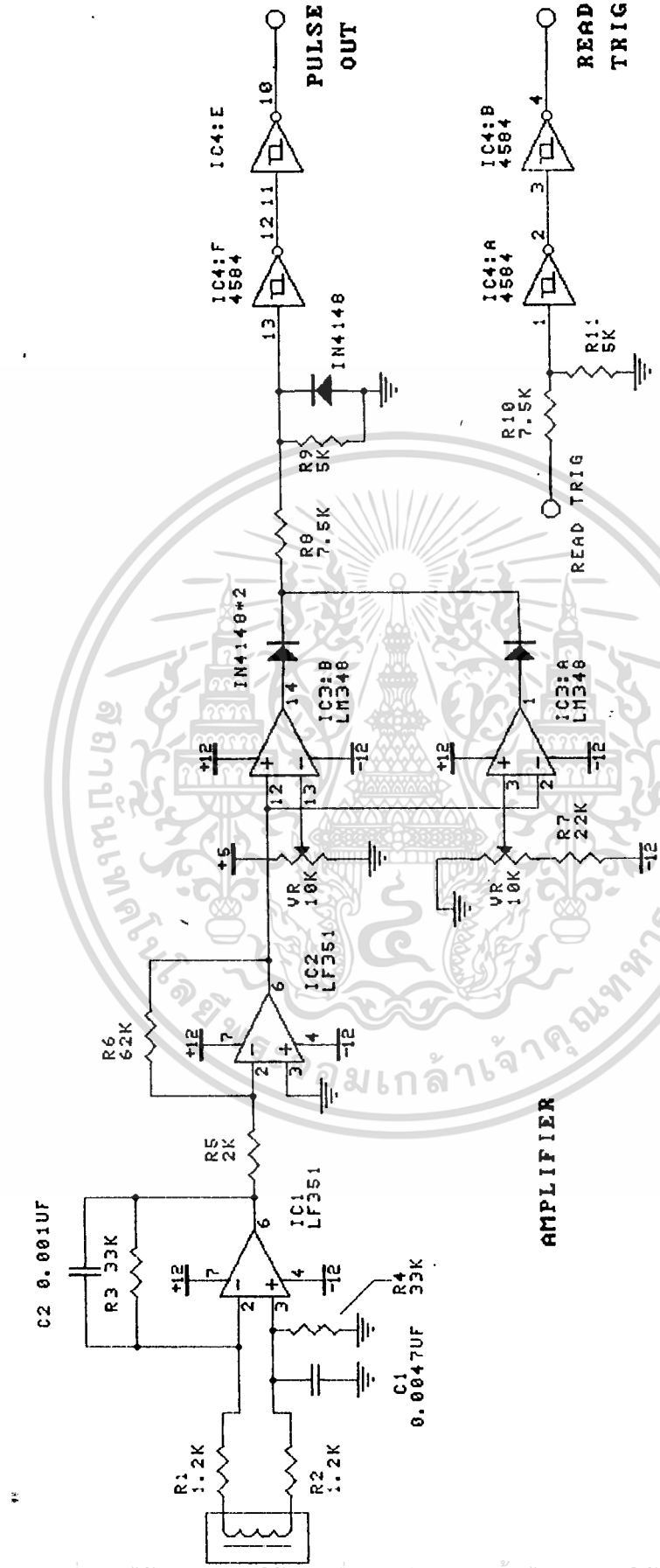
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
034769

2. วงจรขยายและสร้างสัญญาณในการถอดรหัส วงจรนี้แสดงได้ดังรูปที่ 2.6

จากรูปที่ 2.6 วงจรขยายจะใช้ OP-Amp เป็นวงจรขยาย โดยแบ่งเป็น 2 ส่วน โดยในส่วนแรกจะจัดวงจรในลักษณะขยายผลต่าง (Differential Amp) ซึ่งมีอินพุทอิมพีแดนซ์ต่ำ โดยในวงจรนี้จะใช้ OP-amp Low Noise เบอร์ LF351 โดยจะกำหนดค่าเกณฑ์ของวงจรได้จาก R3และ R1 โดย R3 = R4 และ R1 = R2 โดยในวงจรจะได้ค่าเกณฑ์ประมาณ 27 เท่า และมี C1,C2 เป็นตัวลดสัญญาณรบกวน และ VR1 จะใช้ในการปรับค่า OFF Set ของสัญญาณเอาท์พุท ให้มีค่าเป็นศูนย์ เมื่อไม่มีสัญญาณจากหัวเทป ส่วนวงจรส่วนหลังจะใช้วงจรขยายแบบสลับ (Inverting Amp) โดยค่าเกณฑ์จะหาได้จาก R6/R5 ซึ่งจากวงจรจะมีค่าประมาณ 30 เท่า ดังนั้นค่าเกณฑ์ทั้ง 2 ส่วนรวมกันจะได้ประมาณ 800 เท่า ซึ่งจากความแรงของสัญญาณจากหัวเทปที่อ่านได้จะมีค่าประมาณ 5-10 mV ถ้าขยายออกมาก็จะได้ค่าประมาณ 4-8 V ซึ่งเป็นค่าที่จะนำไปใช้ในการตรวจสอบข้อมูลได้ ซึ่งในการออกแบบวงจรขยายนี้จะใช้ Storage Osciloscope เป็นตัวบันทึกสัญญาณในขณะมีการอ่านบัตร และจะมีการเปลี่ยนแปลงค่าอุปกรณ์บางตัวจนได้รูปคลื่นที่ถูกต้องจริง ๆ ซึ่งรูปคลื่นที่ได้จากการ Plot จาก Storage Osciloscope จะแสดงได้ดังรูปที่ 2.7 ก)

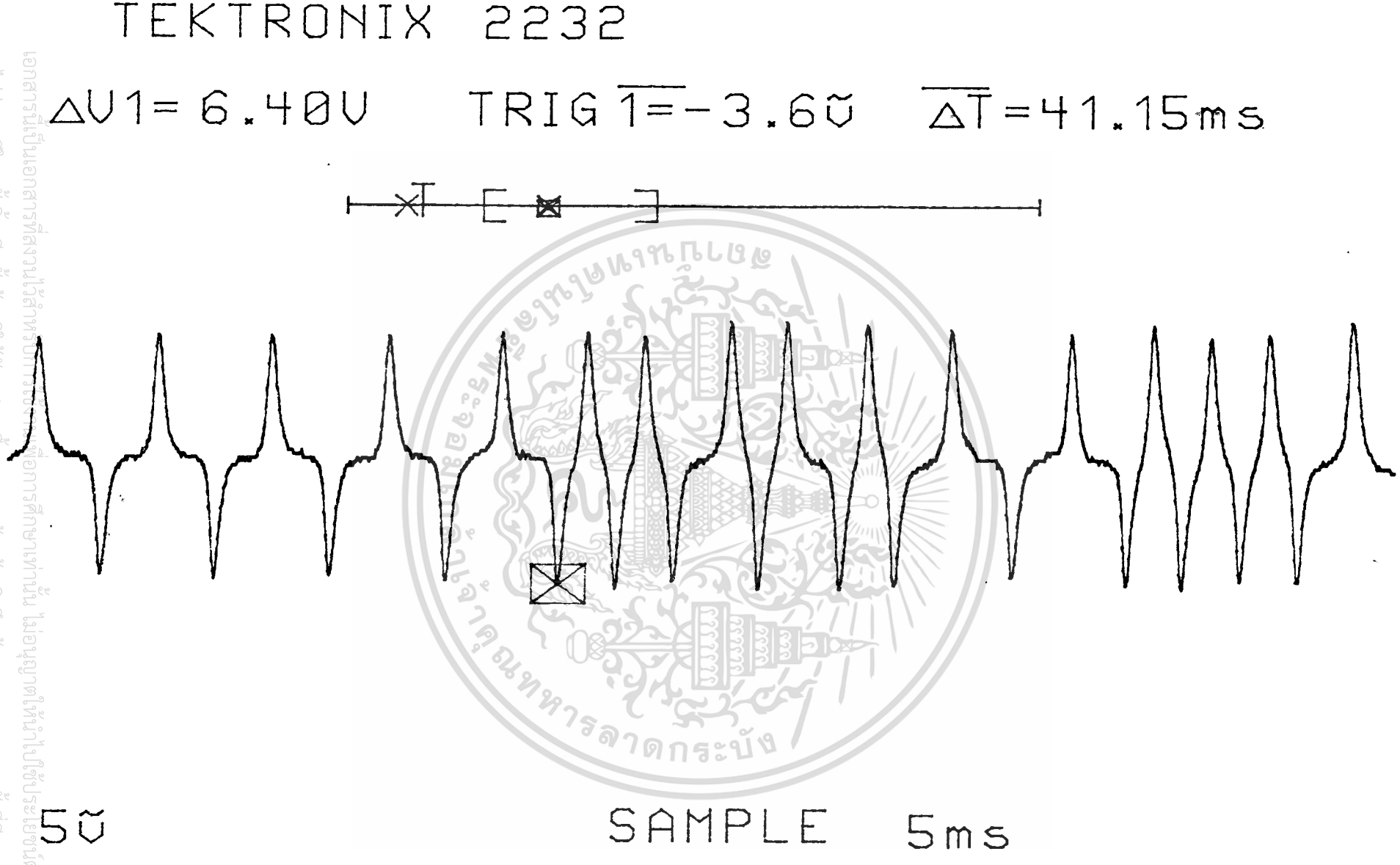
สัญญาณเอาท์พุทจากวงจรขยายจะนำมาทำการเข้าวงจรเปรียบเทียบแรงดัน โดยจะทำการเปรียบเทียบทั้งด้านบวก และด้านลบของสัญญาณ ถ้าสัญญาณมีค่ามากกว่าค่า Reference ทางด้านบวกก็จะได้สัญญาณพัลส์ออกมา แต่ถ้าสัญญาณมีค่าน้อยกว่าค่า Reference ทางด้านลบ ก็จะมีสัญญาณพัลส์ออกมาเหมือนกัน ซึ่งสัญญาณพัลส์ที่ได้นี้แสดงได้ดังรูปที่ 2.7ข) สัญญาณพัลส์ที่ได้นี้จะนำไปเข้าวงจร Schmitt Trigger เพื่อจะทำการเปลี่ยนระดับแรงดันของพัลส์ ให้มีค่าเป็นระดับแรงดันของ TTL (0-5V) เพื่อจะส่งไปเข้าส่วนของ CPU เพื่อทำการ Decode สัญญาณต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

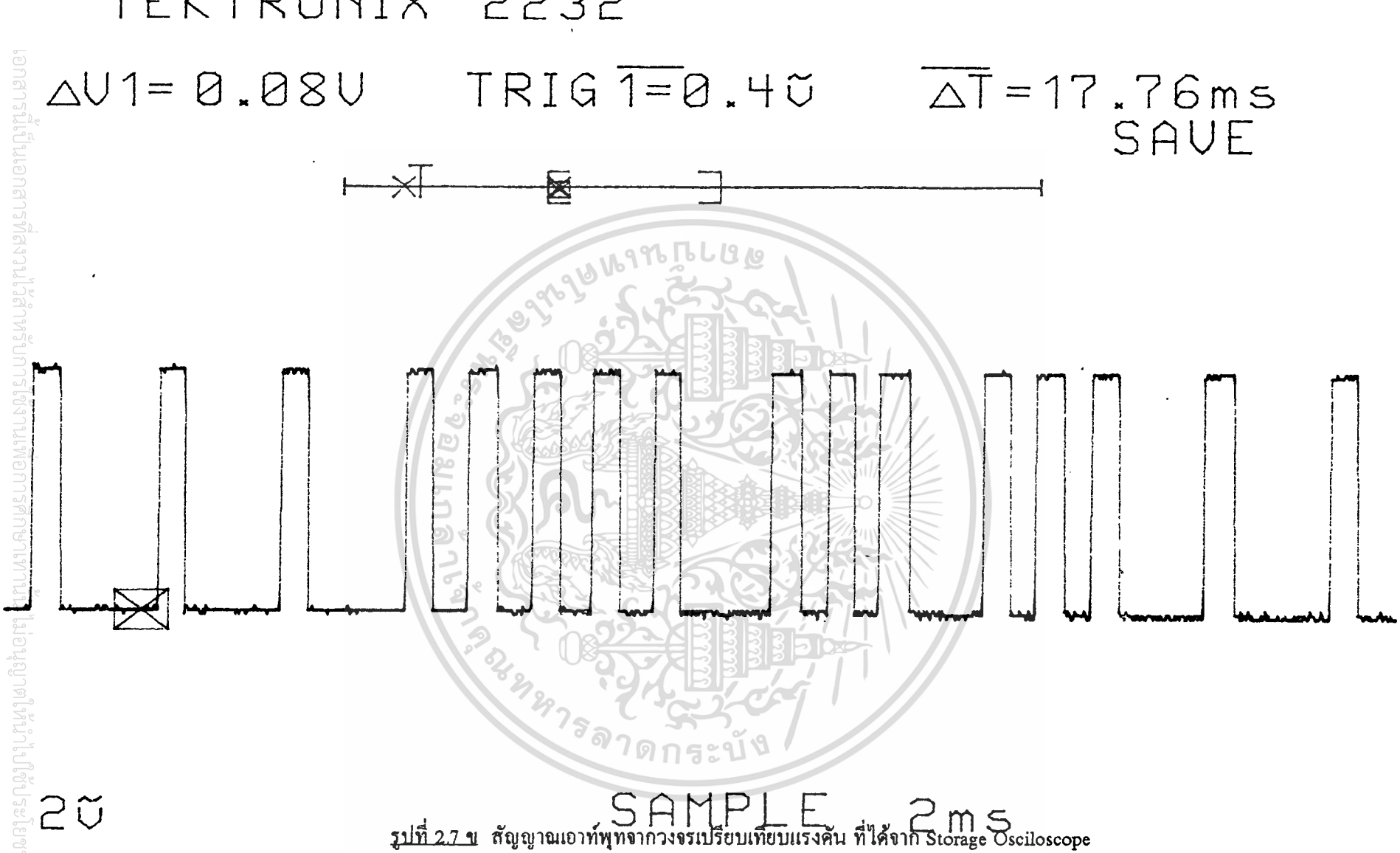


รูปที่ 2.6 วงจรขยายและสร้างสัญญาณในการถอดรหัส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



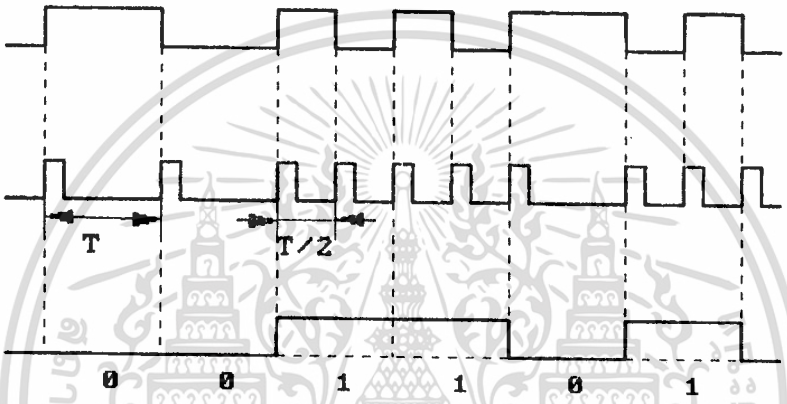
รูปที่ 2.7 ก สัญญาณเอาท์พุทของวงจรขยายสัญญาณ ที่ได้จาก Storage Osciloscope



รูปที่ 2.7 ข สัญญาณเอาท์พุทจากวงจรเปรียบเทียบแรงดัน ที่ได้จาก Storage Osciloscope

3. การถอดรหัสของบัตร

ในการถอดรหัสของบัตรนั้นจะใช้การดูค่าเวลาจากพัลส์หนึ่งไปจนถึงอีกพัลส์หนึ่งซึ่งจะเป็นการบอกว่าข้อมูลของบิทนั้นมีค่าเป็นข้อมูลระดับ "0" หรือ "1" ซึ่งจากในหัวข้อของมาตรฐานของสัญญาณที่บันทึกอยู่ในบัตรแม่เหล็ก จะเห็นได้ว่าบิท "0" จะมีค่าความถี่ต่ำกว่าบิท "1" อยู่ 2 เท่า ถ้ามีการกำหนดให้ค่าเวลาของข้อมูล 1 บิทมีค่าเป็น t จะแสดงลักษณะของสัญญาณ "0" และสัญญาณ "1" ได้ดังรูปที่ 2.8



รูปที่ 2.8 ลักษณะของบิท "0" และบิท "1" จากพัลส์ของวงจรเปรียบเทียบ

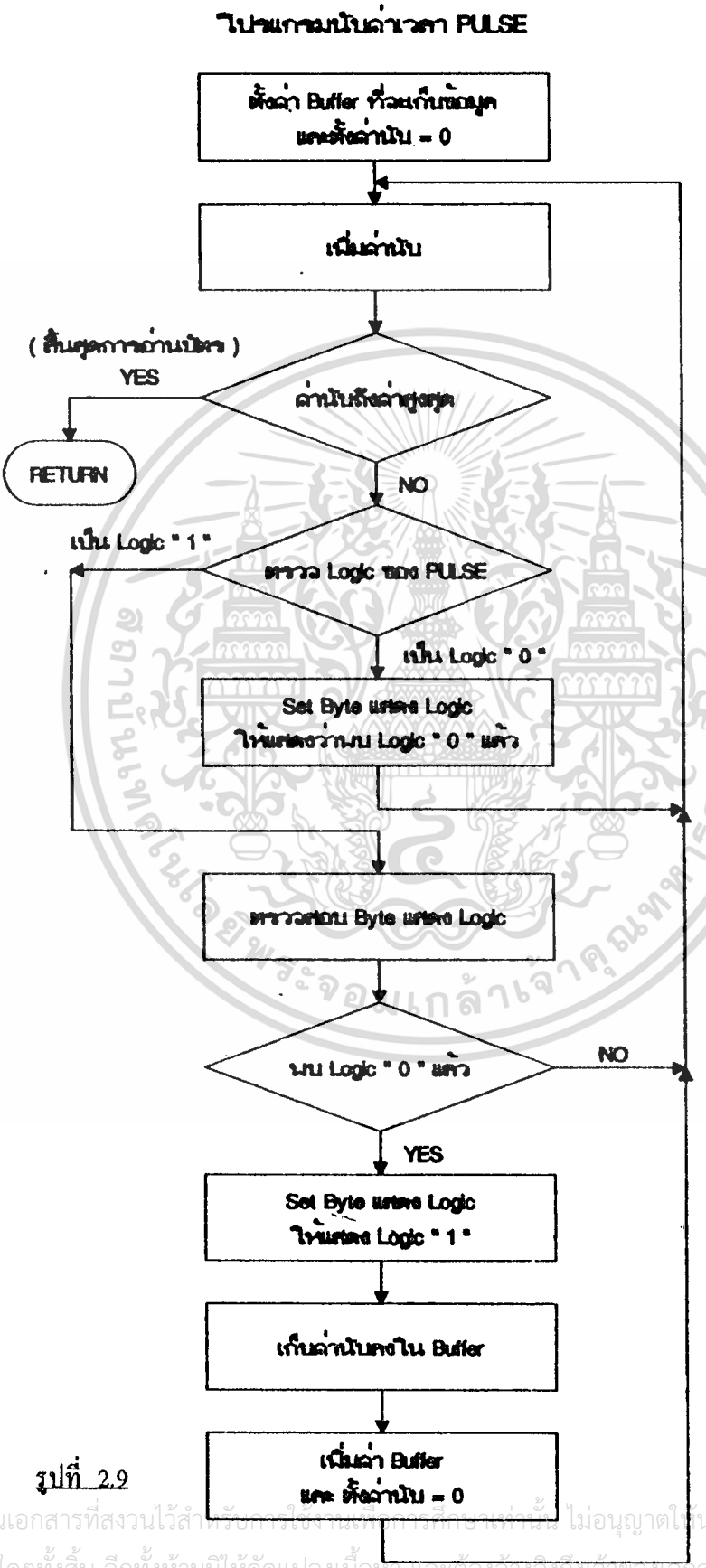
จากรูปที่ 2.8 ลักษณะของบิท "0" จะเกิดจากพัลส์สองลูกที่มีค่าเวลาต่างกันเท่ากับ t ส่วนบิท "1" จะเกิดจากพัลส์สองลูกที่มีค่าเวลาต่างกันเท่ากับ t/2 ติดกัน 2 ครั้ง ซึ่งจากหลักการนี้จะนำไปใช้ในการถอดรหัสของบัตร โดยจะใช้พัลส์นี้ไปเข้า Input Port ของ CPU และจะเขียนโปรแกรมให้ CPU ทำการถอดรหัสออกมา โดยแบ่งส่วนของโปรแกรมได้ 2 ส่วนคือ

3.1 การนับค่าเวลาของพัลส์ โปรแกรมในส่วนนี้แสดงได้ดัง Flow Chart ดังรูปที่ 2.9

จากรูปที่ 2.9 โปรแกรมจะนับค่าเพิ่มขึ้นไปเรื่อย ๆ จนกว่าจะเกิดการเปลี่ยนแปลงจากระดับ "0" ไปเป็นระดับ "1" ซึ่งจะเป็นช่วงที่เกิดพัลส์ลูกใหม่เข้ามา และจะทำการบันทึกค่าที่นับได้ลงใน memory และทำการรีเซ็ทค่านับใหม่ และนับค่าต่อไปเรื่อย ๆ ถ้าเป็นเครื่องอ่านแบบอัตโนมัติก็จะสิ้นสุดโปรแกรมนี้ เมื่อได้รับสัญญาณสิ้นสุดการอ่าน ซึ่งก็คือได้มาจากไมโครสวิทช์ตัวที่ตรวจสอบจุดสิ้นสุดของบัตรนั่นเอง ส่วนเครื่องอ่านแบบใช้มือรูดก็จะสิ้นสุดโปรแกรมนี้เมื่อตรวจสอบว่าค่านับของพัลส์มีค่ามากกว่าความเป็นจริง ซึ่งก็คือเมื่อไม่มีพัลส์เข้ามาอีกแล้ว

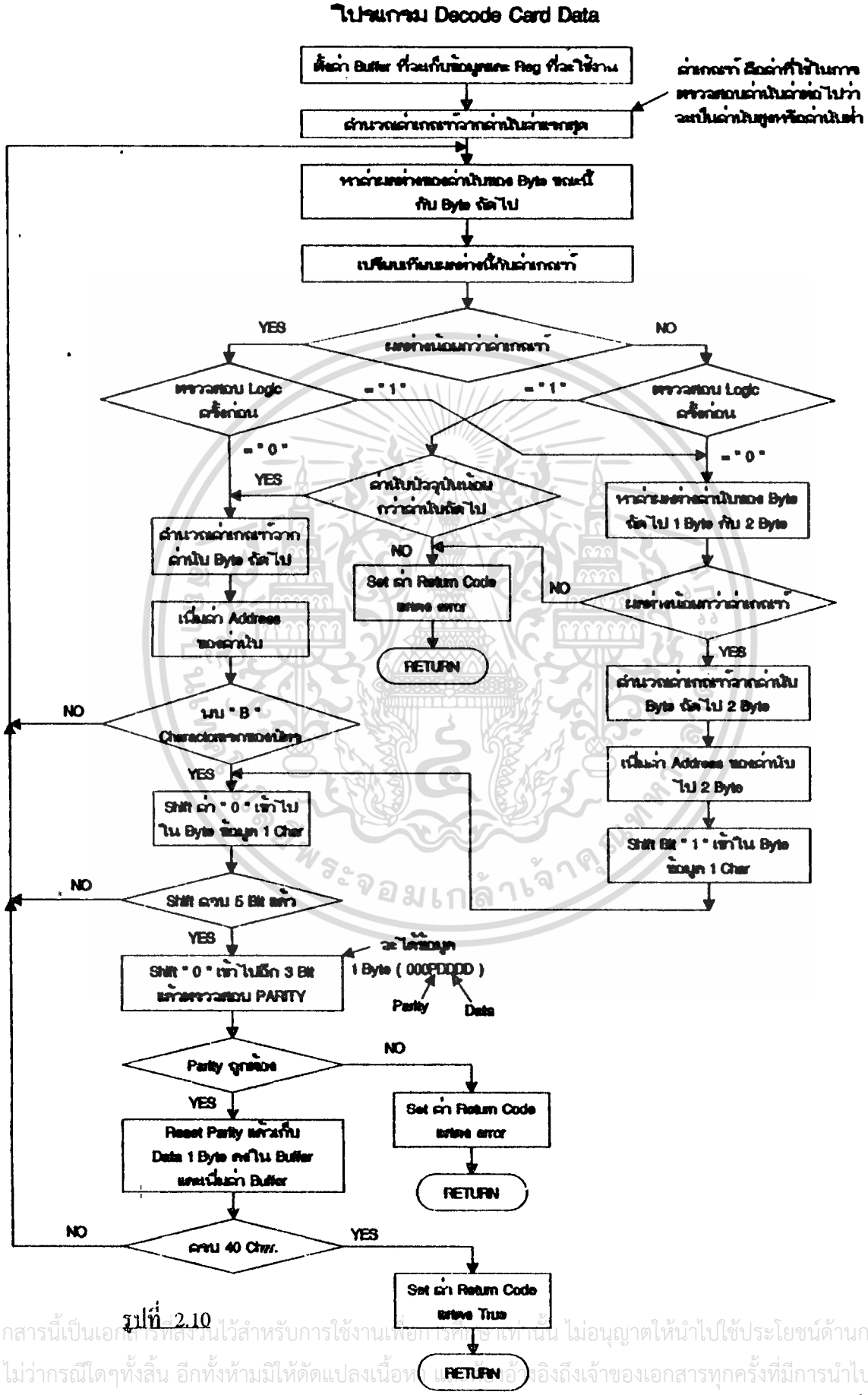
3.2 การแปลงค่าที่นับได้ไปเป็นตัวเลขฐาน 16 ตามมาตรฐานของข้อมูลที่อยู่ในบัตรซึ่งแสดงได้ดัง Flow Chart ดังรูปที่ 2.10

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.9

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.10

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จาก Flow Chart หลักการของโปรแกรมส่วนนี้คือจะกำหนดให้ค่านับครั้งแรกสุดมีค่าเป็น Logic "0" และคำนวณค่านับนี้เป็นค่าเกณฑ์เพื่อที่จะใช้ในการตรวจสอบค่านับครั้งต่อไป ว่าจะมีค่าเป็น Logic อะไร เนื่องจากค่านับที่ได้ ถึงแม้จะเป็น Logic เดียวกัน แต่ค่านับจะไม่เท่ากันเสมอไป ซึ่งจะเกิดจากการความเร็วในการอ่านบัตรไม่เท่ากันตลอด หรือค่าเวลาของพัลส์ของข้อมูลในบัตรไม่เท่ากันตลอดทั้งหมด ดังนั้นจึงต้องมีการกำหนดค่าเกณฑ์ที่จะใช้ในการตัดสินใจว่าค่านับครั้งต่อ ๆ ไป ควรจะมีค่าเป็น Logic อะไร และจะต้องมีการคำนวณค่าเกณฑ์ เพื่อใช้ในการตัดสินใจจากค่านับที่ติดกัน ครั้งต่อครั้ง เนื่องจากค่านับที่ติดกันซึ่งเกิดจากการอ่านบัตรที่ในเวลาใกล้เคียงกัน ย่อมเกิดความเปลี่ยนแปลงของค่านับมีค่าน้อย โอกาสที่จะเกิดการผิดพลาดก็มีได้น้อย แต่ถ้าใช้ค่าเกณฑ์ครั้งแรกในการตัดสินใจกับค่านับอื่นตลอดทั้งหมด ย่อมจะเกิดการผิดพลาดได้อย่างแน่นอน ซึ่งถ้าเป็นเครื่องอ่านแบบใช้มือรูดด้วยแล้ว จะยิ่งเกิดความผิดพลาดขึ้นได้แน่นอน เพราะความเร็วในการรูดบัตรของคนจะมีค่าไม่เท่ากันตลอดการรูด คือมีเร็วบ้าง ช้าบ้าง

ในตอนแรกของข้อมูลที่อ่านได้ จะเป็นสัญญาณ Sync ซึ่งก็คือ มีค่าเป็น Logic "0" ที่มีจำนวนหลาย ๆ บิทติดต่อกัน โปรแกรมก็จะทำการหาข้อมูลไปจนพบ Logic "1" และจะเริ่มทำการเรียงข้อมูลที่ได้ต่อไปจนครบ 5 บิท จากนั้นก็จะตรวจสอบ Parity Bit และตรวจสอบตัวเลขที่ได้ว่าเป็นตัวเริ่มข้อมูลหรือยัง (B) ถ้าไม่ใช่ก็จะทำการแจ้งผลว่า Eror แต่ถ้าถูกต้องก็จะทำการเรียงข้อมูลต่อไปจนพบตัวเลขแสดงจุดสิ้นสุดของข้อมูล (F) จากนั้นก็จะทำการตรวจสอบค่า CRC ของข้อมูลทั้งหมดโดยจะใช้วิธีการตรวจสอบค่าบิทจากข้อมูลทุกตัว เป็น Parity ซึ่งก็คือ นำบิท 0 ของข้อมูลทั้งหมดมาทำการคำนวณหา Parity เป็นค่า CRC บิท 0 และบิท 1,2,3 และรวมถึง Parity Bit ของแต่ละตัวเลขมาคำนวณหา Parity เหมือนบิท 0 ทั้งหมด ซึ่งก็จะได้ค่า CRC จำนวน 5 บิท ไปใช้เทียบกับค่า CRC ที่อยู่ในบัตร ถ้ามีค่าตรงกันก็แสดงว่าค่าที่อ่านได้มีค่าถูกต้อง แต่ถ้าไม่ตรงกัน ก็แสดงว่าค่าที่อ่านได้มีค่าผิดพลาด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

การติดต่อสื่อสารข้อมูล

จากโครงสร้างของระบบที่อธิบายอย่างคร่าว ๆ ในบทที่ 1 จะมีส่วนประกอบสำคัญคือตัวคอมพิวเตอร์, Center และ Terminal ดังนั้นในการติดต่อสื่อสารข้อมูลจะเกิดขึ้นระหว่างคอมพิวเตอร์กับ Center และระหว่าง Center กับ Terminal

การติดต่อระหว่าง คอมพิวเตอร์กับ Center

ข้อมูลที่มีการติดต่อกันในส่วนนี้ คือเครื่อง Center จะทำการส่งค่าเวลาเข้าออกทำงานของพนักงานไปให้คอมพิวเตอร์ เพื่อเก็บข้อมูลเวลาเข้าออกทำงานของพนักงานทุกวัน โดยมาตรฐานที่ใช้ในการสื่อสารข้อมูลในส่วนนี้จะใช้มาตรฐาน RS-232 ซึ่งเป็นการรับส่งข้อมูลแบบ Full Duplex แบบใช้กราวนด์ร่วม และแทนระดับ Logic 1 ด้วยแรงดันบวกประมาณ 15 โวลท์ และระดับ Logic 0 ด้วยค่าแรงดันลบประมาณ -15 โวลท์ ซึ่งในมาตรฐานนี้เป็นมาตรฐานที่หาข้อมูลประกอบได้ง่าย และยังเป็นมาตรฐานที่รู้จักกันทั่วไป ดังนั้นจึงไม่ขอกล่าวอย่างละเอียดในส่วนนี้

การติดต่อระหว่าง Center กับ Terminal

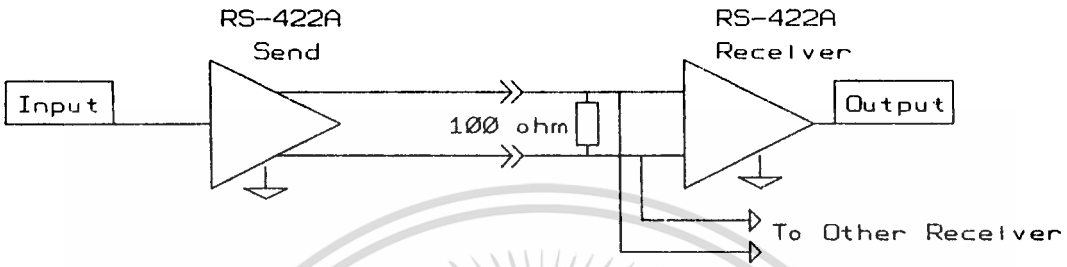
การติดต่อกันในส่วนนี้ เกิดจาก Center ไปทำการติดต่อกับ Terminal ซึ่งมีจำนวนทั้งหมด 16 ตัว โดยจะใช้มาตรฐานการสื่อสาร RS-485 ซึ่งเป็นการติดต่อแบบ Half Duplex โดยผ่านทางสายสัญญาณเพียงคู่เดียว และไม่ใช้กราวนด์ร่วม โดยจะมีรายละเอียดของมาตรฐานดังนี้

ลักษณะของ RS-485

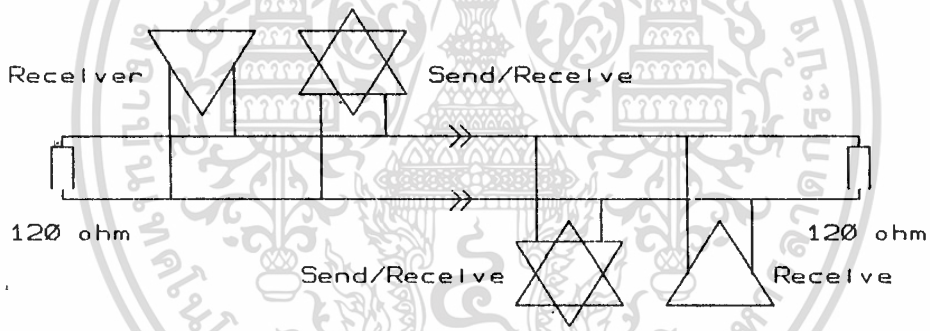
RS-485 เป็นมาตราฐานการสื่อสารข้อมูลแบบสมดุลย์ที่ได้พัฒนามาจาก RS 422-A เพื่อให้ตัวรับและตัวส่งสามารถรับส่งได้จำนวนมากคู่ สามารถใช้คู่สายสัญญาณรับส่งร่วมกันได้ (Multipoint Multiple Drivers And Receivers) ซึ่งในกรณีของ RS-422-A คู่สายสัญญาณรับส่งหนึ่งคู่ จะมีตัวรับได้ไม่เกิน 10 ชุด และมีตัวส่งได้เพียงหนึ่งชุด แต่ในกรณีของ RS-485 สามารถใช้ตัวรับ 32 ชุด และตัวส่ง 32 ชุดได้ โดยใช้สายสัญญาณร่วมกันหนึ่งคู่ โดยทั่ว ๆ ไป RS-485 มีคุณเลักษณะเฉพาะทางไฟฟ้าของตัวรับและตัวส่งคล้ายกับ ตัวรับและตัวส่งของ RS-422-A และไม่จำกัดรูปแบบของโปรโตคอลที่จะนำมา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใช้งานกับระบบที่พัฒนาขึ้น โดยขึ้นอยู่กับผู้พัฒนาระบบเองว่าจะเลือกใช้โปรโตคอลแบบไหนมาใช้งาน นอกจากนี้ตัวรับและตัวส่งมีราคาไม่สูงทำให้ RS-485 ถูกนำมาประยุกต์ใช้งานในระบบการสื่อสารข้อมูลแบบอนุกรมอย่างแพร่หลาย



รูปที่ 3.1 ก) แสดงการเชื่อมต่อของ RS-422-A



รูปที่ 3.1 ข) แสดงการเชื่อมต่อของ RS-485

จากรูปที่ 3.1 เป็นลักษณะการเชื่อมต่อของ RS-422-A เปรียบเทียบกับระบบของ RS-485 โดย RS-422-A ในหนึ่งคู่สายสัญญาณรับส่งจะมีตัวส่งเพียงชุดเดียว ส่วนตัวรับมีได้สูงสุดไม่เกิน 10 ชุด พร้อมกับตัวต้านทาน 100 โอห์ม ต่อคร่อมระหว่างคู่สายสัญญาณรับส่ง จากลักษณะดังกล่าว การสื่อสารข้อมูลจึงเป็นแบบทิศทางเดียว ทำให้เกิดปัญหาในการสื่อสารข้อมูลจากตัวรับกลับมาที่ตัวส่ง เพราะจะต้องเพิ่มคู่สายสัญญาณรับส่งอีกหนึ่งคู่สาย เพื่อทำให้สามารถสื่อสารข้อมูลแบบสองทิศทางได้ จึงได้มีการพัฒนามาเป็น RS-485 ซึ่งสามารถรับและส่งได้ภายในคู่สายสัญญาณเพียงคู่เดียว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คุณสมบัติเฉพาะของ RS-485 ที่แตกต่างจาก RS-422-A

1. คุณลักษณะเฉพาะของตัวส่ง RS-485

- ตัวส่งหนึ่งตัวสามารถขับโหลดได้ถึง 32 ชุด (โหลดหนึ่งชุดประกอบด้วยตัวส่ง 1 ตัว และตัวรับ 1 ตัว) และ ค่าความต้านทานรวมที่ต่อคร่อมคู่สายสัญญาณมีค่า 60 โอห์ม หรือมากกว่า

- เอาท์พุทของตัวส่งในสภาวะ OFF มีกระแสรั่วไหลไม่เกิน 100 uA ในช่วงแรงดันไฟฟ้าโหมดร่วมระหว่างค่า -7 โวลท์ ถึง 7 โวลท์

- เอาท์พุทของตัวส่งให้แรงดันไฟฟ้าเอาท์พุท 1.5 โวลท์ ถึง 5 โวลท์ ในช่วงแรงดันไฟฟ้าโหมดร่วมระหว่างค่า -7 โวลท์ ถึง 12 โวลท์

- ตัวส่งมีวงจรป้องกันตัวเองที่ส่วนเอาท์พุท ในกรณีที่ตัวส่งหลาย ๆ ตัวส่งข้อมูลออกมาพร้อม ๆ กัน

2. คุณลักษณะเฉพาะของตัวรับ RS-485

- ค่าความต้านทานที่อินพุทมีค่าสูง โดยมีค่าไม่น้อยกว่า 12 กิโลโอห์ม

- ตัวรับมีค่าแรงดันไฟฟ้าอินพุทโหมดร่วม ระหว่างค่า -7 โวลท์ถึง 12 โวลท์

- ตัวรับสามารถตอบสนองต่อสัญญาณที่แตกต่างจากสัญญาณโหมดร่วมได้ ± 200 mV (น้อยที่สุด)

3. คุณลักษณะเฉพาะของคู่สายสัญญาณ RS-485

- คู่สายสัญญาณรับส่ง ควรพันสลับกันเป็นเกลียวเพื่อลดทอนสัญญาณรบกวน

4. ความหมายของยูนิตโหลด (Unit Load)

เป็นจำนวนมากที่สุดของตัวรับและตัวส่ง ที่สามารถใช้งานบนคู่สายสัญญาณรับส่งหนึ่งคู่ โดยจะขึ้นอยู่กับค่า Unit Load (U.L.) ซึ่ง RS-485 ยอมรับได้ที่ 32 Unit Load ต่อคู่สายสัญญาณหนึ่งคู่

ค่า 1 U.L. ถูกนิยามไว้ดังนี้ (ในกรณีที่มีปัญหามากที่สุด)

" เป็นโหลดที่ใช้กระแส 1 มิลลิแอมป์ ที่แรงดันไฟฟ้าโหมดร่วม 12 โวลท์ ซึ่งโหลดนี้จะประกอบด้วย ตัวส่ง และ/หรือ ตัวรับ แต่ไม่รวมค่าความต้านทานที่เกิดจากตัวต้านทานที่ต่อคร่อมคู่สายสัญญาณรับส่ง "

คุณสมบัติของคู่ตัวรับ-ส่ง (Transceivers) RS-485

คู่ตัวรับ-ส่ง (Transceivers) เป็นอุปกรณ์ที่มีทั้งตัวรับ และตัวส่งอยู่ในชิพเดียวกัน เพื่อให้เกิดความสะดวกในการใช้งาน และทำให้ระบบมีขนาดเล็กลง คู่ตัวรับ-ส่ง ของ RS-485 มีอยู่หลายเบอร์ ได้แก่ SN75176B ,SN75177B ,SN75178B และ SN75179B

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

<u>คุณลักษณะเฉพาะของคู่ตัวรับ-ส่ง (Transceivers)</u>

- ตามมาตราฐาน RS-485 , RS-422-A , CCITT V.11 และ X.27
- เอาท์พุทของตัวส่งเป็นแบบ 3-State ยกเว้น SN75179B
- เอาท์พุทตัวส่งสามารถขับกระแสได้สูง 60 มิลลิแอมป์
- Thermal Shutdown Protection
- ค่าความต้านทานอินพุทของตัวรับ 20 กิโลโอห์ม (น้อยที่สุด)
- ตัวรับมีค่า Input Sensitivity ± 200 มิลลิโวลท์
- คู่ตัวรับมีค่า Input Hysteresis 50 มิลลิโวลท์
- ใช้ไฟเลี้ยง 5 โวลท์

<u>วิธีการติดต่อสื่อสาร ผ่านทาง RS-485</u>

เนื่องจากมาตราฐาน RS-485 นี้เป็นลักษณะการการติดต่อกันในลักษณะ Half Duplex คือ จะมีตัวส่งได้เพียง 1 ตัว ในช่วงเวลาหนึ่งเท่านั้น ซึ่งก็คือจะมีอุปกรณ์ที่ต่ออยู่ในระบบมีการส่งข้อมูลออกมาพร้อมกันไม่ได้ แต่ในส่วนของการรับจะรับพร้อมกันได้หมดทุกตัว ดังนั้นจึงต้องออกแบบโปรโตคอลเพื่อจะให้ติดต่อกันได้อย่างถูกต้อง และเหมาะสมกับงานนี้

หน้าที่หลักของโปรโตคอลที่จะใช้ในการออกแบบระบบการสื่อสารควรจะมีดังต่อไปนี้

- สร้างกรอบข่าวสาร (framing) หน้าที่คือการแบ่งข่าวสารออกเป็นบล็อก ๆ ซึ่งประกอบไปด้วย ส่วนเนื้อหา (text) และส่วนควบคุม โดยในการติดต่อระหว่าง Center กับ Terminal ข้อมูลที่ส่งระหว่างกัน จะเป็นข้อมูลที่ไม่มีความยาวมากนัก เช่นเป็นข้อมูลของบัตรหรือเป็นคำสั่งต่าง ๆ ซึ่งมีข้อมูลที่มีขนาดสั้น ๆ

- การควบคุมการผิดพลาด มีหน้าที่คือ ตรวจจับความผิดพลาด เช่น การใช้ CRC 16 บิท และรวมไปถึงการแก้ไขเมื่อมีการตรวจพบความผิดพลาด เช่น การร้องขอให้ส่งสัญญาณข่าวสารกลับมาใหม่

- การควบคุมลำดับในการส่งข้อมูล มีหน้าที่คือ จัดลำดับในการส่งข้อมูลระหว่างอุปกรณ์ให้เป็นไปอย่างถูกต้อง เช่นจะให้อุปกรณ์ตัวหนึ่ง ๆ ส่งข้อมูลเมื่อใดเป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วิธีการติดต่อสื่อสารในระบบนี้ จะใช้ตัว Center เป็นตัวควบคุมการรับส่งข้อมูลภายในระบบทั้งหมด (Host) โดยในขั้นแรกจะต้องมาดูรูปแบบของข้อมูลในการติดต่อสื่อสารเสียก่อน ซึ่งมาตราฐานที่ใช้ในการติดต่อผ่านระบบของ RS-485 นี้จะมีรูปแบบดังนี้

Sync	Address	Command	Data	CRC	EOT
(16H)	(80-8F)	(D0-EFH)	(20-7FH)		(04H)

Sync เป็น Byte ที่แสดงว่าเริ่มส่งข้อมูล (หรือ เป็น Byte แรก ของข้อมูลใน บล็อค)

Address เป็น Byte ที่แสดง Terminal ตัวที่ Center จะทำการติดต่อ หรือ แสดง Terminal ตัวที่ส่งข้อมูลมาให้ Center

Command เป็น คำสั่งที่ให้ทำงาน ต่าง ๆ.

Data เป็น ข้อมูลที่ส่ง ซึ่งในบางคำสั่ง อาจไม่ต้องใช้ Data ก็ได้

CRC เป็นค่าผลรวมของ ข้อมูลทั้งบล็อก ที่ใช้ในการตรวจสอบความถูกต้องของข้อมูลทั้งหมด ภายในบล็อค

EOT เป็น Byte ที่แสดงการสิ้นสุดการส่งข้อมูล

ในส่วนของ Data นั้น จะมี รหัส STX เป็นตัวเริ่มบล็อก ของ Data และมี ETX เป็นตัวสิ้นสุดของบล็อก ของ Data โดย รหัสที่ใช้ จะเป็นรหัส ASCII ทั้งหมด และไม่ซ้ำกับรหัสที่เป็นรหัสที่ใช้ควบคุม

โดยในระบบนี้จะออกแบบให้ตัว Center เป็นตัวควบคุมการรับ-ส่งข้อมูลภายในระบบซึ่งก็คือจะต้องไปทำการติดต่อกับส่วนของ Terminal (16 ตัว) ถ้า Center ต้องการติดต่อกับ Terminal ตัวใดก็จะส่งข้อมูลผ่านไปทาง RS-485 นี้ โดยจะมี Byte ของข้อมูล 1 Byte แสดง Terminal ตัวที่จะทำการติดต่อด้วย (Address) ถ้า Terminal ตัวนั้นตรวจพบว่าเป็น Address ของมันเอง ก็จะทำการส่งข้อมูลกลับมาให้ยัง Center (ผ่านทางสายเดียวกัน) ส่วน Terminal ตัวที่ไม่ใช่ที่ Center ต้องการจะติดต่อ ก็จะไม่สนใจข้อมูลที่ส่งมา

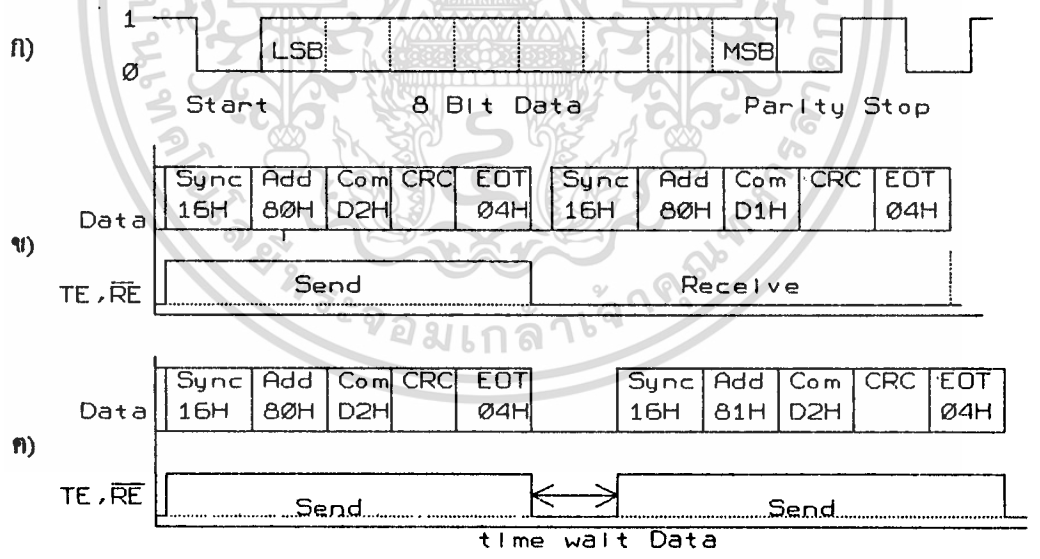
ซึ่งจะเห็นได้ว่า Terminal จะยังไม่ส่งข้อมูลออกไปถ้ายังไม่ได้รับอนุญาติจาก Center ก่อน (คำสั่ง ที่ให้ terminal ส่งข้อมูลออกมา) โดย Terminal จะทำการตรวจสอบข้อมูลที่รับได้ ทีละไบต์ ซึ่งจะไม่ใช้วิธีการเก็บลงบัฟเฟอร์ ก่อนแล้วค่อยมาตรวจสอบภายหลัง ซึ่งถ้าใช้การเก็บลงบัฟเฟอร์ก่อนแล้วค่อยมาตรวจสอบนี้ จะทำให้ จังหวะการติดต่อกัน ระหว่าง Center กับ Terminal นี้ เกิดการผิดพลาด อาจจะเกิดข้อมูลที่ส่งออกมาพร้อมกัน ซึ่งถ้าใช้การตรวจสอบทีละไบต์ จะแก้ปัญหานี้ได้ เพราะจะมีการตรวจสอบจังหวะการทำงานทุก ๆ ไบต์ เช่น เมื่อ Center ส่งข้อมูลไป Poll Terminal ตัวหนึ่ง เมื่อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สิ้นสุดการส่งข้อมูลของบล็อคแล้ว ก็จะทำการ รอข้อมูลกลับจาก Terminal โดยจะมีช่วงเวลารอข้อมูลมากกว่าเวลาของข้อมูล 1 ไบต์ เล็กน้อย ถ้าในช่วงเวลานี้ ไม่มีข้อมูลตอบรับกลับมา ก็จะถือว่า ไม่มีการติดต่อจาก Terminal และจะไปทำการติดต่อกับ Teminal ตัวอื่น ต่อไป

ซึ่งจากรูปแบบของการส่งข้อมูล ของ RS-485 นี้ จะเห็นว่า ในส่วนของรหัสของ Control Charactors (Sync, EOT เป็นต้น) จะเป็นค่า HEX Code ที่มีรหัสไม่ซ้ำกับส่วนอื่น ๆ เช่นในส่วนของ Address, Command, Data, CRC ทั้งนี้ ก็เพื่อต้องการจะแยก Control Charactors ออกจากส่วนอื่นทั้งหมด เนื่องจาก ในระบบการติดต่อสื่อสาร ระบบนี้ อุปกรณ์ในระบบ (Terminal) จะไม่ได้รับข้อมูลจาก Center อยู่ตลอดเวลา ซึ่ง ก็คือในส่วนของ Termimal จะต้องมีการ อ่านบัตรแม่เหล็ก ที่จะใช้ในการเข้าออก หรือบันทึกเวลาในการทำงาน ซึ่งในช่วงเวลาดังกล่าว Terminal จะไม่รับข้อมูลจาก Center ซึ่งเมื่อ Terminal เริ่มรับข้อมูลจาก Center ใหม่อีก ใน Byte แรกของข้อมูลที่ Terminal รับได้ในตอนนี้ อาจจะไม่ใช่ Byte เริ่มต้นของ บล็อคข้อมูล (Byte Sync) ก็ได้ ดังนั้น จึงต้องมีการออก แบบให้ Control Charactors (โดยเฉพาะ Sync และ EOT) มีค่า ที่ไม่ซ้ำกับข้อมูลในส่วนอื่นของบล็อคข้อมูล ทั้งหมดด้วย

ตัวอย่างการส่งข้อมูล เมื่อ Center ส่งข้อมูลไป Poll Terminal #1 แสดงได้ดังรูปที่ 3.2



รูปที่ 3.2 รูปแบบสัญญาณเมื่อมีการติดต่อระหว่าง Center กับ Terminal #1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 3.2 ก) เป็นลักษณะของสัญญาณแบบอนุกรมที่ใช้ในการติดต่อ 1 ตัวอักษร

รูปที่ 3.2 ข) แสดงเมื่อมีการติดต่อกลับมาจาก Terminal #1 ซึ่งก็คือ Center จะทำการ Poll ไปที่ Terminal #1 โดยการตั้งค่า Address ให้มีค่าเท่ากับ 80H ซึ่งมีค่าตรงกับTerminal #1 และในส่วนของ Command จะมีค่าเป็น D2H ซึ่งเป็นคำสั่งที่ใช้ Poll Terminal ให้ Terminal ตอบรับกลับมาได้ เมื่อ Terminal พร้อมจะตอบรับ ก็จะส่งข้อมูลกลับมาโดยมีค่า Address เป็นของตัวเอง และ Command กลับไปให้ Center

รูปที่ 3.2 ค) แสดงเมื่อไม่มีการติดต่อกลับมาจาก Terminal #1 ซึ่งก็คือเมื่อ Center ทำการ Poll ไปที่ Terminal #1 และยังไม่มีการตอบรับกลับมาจาก Terminal #1 หรือยังไม่มีสัญญาณผ่านมาทางสายสัญญาณจนครบเวลา Time wait data ดังรูป Center ก็จะไปทำการ Poll Terminal ตัวต่อไป

<u>คำสั่ง (command) ที่ Center ส่งไปที่ Terminal</u>

1. D0H เป็นข้อมูลของค่าเวลาในขณะนั้น โดยจะส่งไปให้ Terminal ทุกตัว โดยในส่วนของ Field Address จะมีค่าเป็น A0H ซึ่งหมายถึง Terminal ทุกตัว มีรูปแบบในส่วนของ Field Data ดังนี้

STX	0EH	10 Hour	1Hour	10 minute	1 minute	ETX

2. D2H เป็นคำสั่ง Poll เพื่อให้ Terminal ตัวที่มีค่าตรงกับ Field Address ตอบรับกลับมา โดยคำสั่งนี้ไม่มีส่วนของ Field Data

3. D4H เป็นคำสั่งให้ Terminal ตัวที่มีค่าตรงกับ Field Address ทำการตอบรับกลับมา และรอรับข้อมูลจาก Center โดยคำสั่งนี้ไม่มีส่วนของ Field Data

4. D6H เป็นคำสั่งส่ง Mode ไปให้ Terminal ตัวที่มีค่าตรงกับ Field Address โดยคำสั่งนี้จะมีส่วนของ Field Data ดังนี้

STX	0FH	Mode	ETX

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดย Mode จะมี 3 Mode คือ

- 30H มีค่าเท่ากับ Mode 1
- 31H มีค่าเท่ากับ Mode 2
- 32H มีค่าเท่ากับ Mode 3

5. D8H เป็นคำสั่งตอบรับการตรวจสอบการเข้าออก ไปให้ Terminal ตัวที่มีค่าตรงกับค่าใน Field Address ปฏิบัติตามค่า Return Code ที่ส่งไปให้ โดยมีรูปแบบของ Field Data ดังนี้

STX	10H	Return Code	ETX

โดย Return Code มีดังนี้

- 30H มีค่าเท่ากับ Pass OK
- 31H มีค่าเท่ากับ No Pass
- 32H มีค่าเท่ากับ Code Missing
- 33H มีค่าเท่ากับ Pass and Work in OK
- 34H มีค่าเท่ากับ Work out OK
- 35H มีค่าเท่ากับ Work out missing

คำสั่ง (Command) ที่ Terminal ส่งไปให้ Center

1. D1H เมื่อ Terminal ต้องการให้ Center ส่ง Mode มาให้ คำสั่งนี้ไม่มีส่วนของ Field Data

2. D3H Terminal ส่งข้อมูลของบัตร และรหัสไปให้ Center เพื่อทำการตรวจสอบการเข้าออก และให้ส่งผลที่ตรวจสอบได้กลับมาให้ คำสั่งนี้มีรูปแบบของ Field Data ดังนี้

STX	Card Data	1DH	IN,OUT	Code	ETX

3. D5H Terminal อยู่ในสถานะว่าง คือไม่มีข้อมูลจะส่งให้ Center คำสั่งนี้ไม่มีส่วนของ Field Data

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

โครงสร้างของระบบ

จากรูปที่ 1.1 แสดงโครงสร้างของระบบ จะเห็นได้ว่าระบบจะประกอบไปด้วย ตัวเครื่องคอมพิวเตอร์ และส่วนควบคุม การเข้าออก โดยตัวคอมพิวเตอร์จะเก็บข้อมูลของ เวลาที่พนักงานเข้าออกทำงาน ในทุก ๆ วัน และคำนวณเป็นเงินเดือนออกมา เมื่อสิ้นสุดวันสุดท้ายของการทำงานในแต่ละเดือน และส่วนควบคุมการเข้า-ออกจะประกอบไปด้วย Center ซึ่งเป็นศูนย์กลางของระบบ และ Terminal ซึ่งเป็น ส่วนย่อย ที่ต่ออยู่ในระบบ โดย Center จะเป็นส่วนที่เก็บข้อมูลของ บัตร, รหัส ของพนักงาน และบุคคลทั้งหมดที่จะผ่านเข้า-ออก และ Terminal จะเป็นส่วนที่ตรวจสอบการเข้าออก ตามจุดต่าง ๆ ดังนั้น Terminal จะมีจำนวนหลายตัว และติดตั้งอยู่ตามจุดต่าง ๆ ที่จะมีการเข้าออก และจะบันทึกเวลาเข้า-ออกทำงานของพนักงาน หรือ ผ่านเข้าไปตามธรรมดา (ไม่บันทึกเวลา) และ Center จะทำการส่งข้อมูลของเวลาเข้า-ออก ไปให้คอมพิวเตอร์ทุกวัน โดยในการติดต่อระหว่าง Center กับคอมพิวเตอร์ เพื่อส่งข้อมูลค่าเวลาเข้า-ออกในการทำงานของพนักงาน ให้คอมพิวเตอร์ จะใช้มาตรฐานการสื่อสารข้อมูล RS-232 ซึ่งเป็นการสื่อสารแบบ Full Duplex คือส่งและรับ ได้ในเวลาเดียวกัน ส่วนการติดต่อระหว่าง Center กับ Terminal จะใช้มาตรฐานการสื่อสารข้อมูล RS-485 ซึ่งเป็นการสื่อสารแบบ Half Duplex คือ ส่งและรับได้ แต่จะต้องผลัดกัน โดยทั้ง 2 มาตรฐานนี้และ วิธีการติดต่อระหว่าง Center และ Terminal ได้ทำการแสดงรายละเอียดอยู่ในบทที่ 3 เรื่องการติดต่อสื่อสารข้อมูล

<u>การทำงานของระบบ</u>

Center จะเป็นส่วนที่เป็นศูนย์กลางของระบบ จะทำการติดต่อกับ Terminal โดยใช้วิธีการ Poll ไปที่ Terminal ทั้ง 16 ตัว ซึ่งวิธีการส่งข้อมูลเพื่อทำการ Poll แสดงอยู่ในบทที่ 3 ในเรื่องการติดต่อสื่อสารข้อมูล

ในการเข้าออกที่ Terminal จะต้องมีการรูดบัตรเพื่อที่จะเข้าออก และจะมีการกดรหัสหรือไม่กดรหัส ขึ้นอยู่กับ Mode ของการใช้งานของ Terminal ในขณะนั้น ซึ่งมีอยู่ 3 Mode ดังนี้

- Mode 1 ไม่ต้องมีการกดรหัสผ่าน
- Mode 2 ต้องมีการกดรหัสผ่านทุกครั้ง
- Mode 3 ในการเข้าเมื่อประตูปิดอยู่ จะต้องมีการกดรหัสผ่าน แต่ถ้าเข้าในเวลาที่ประตูยังเปิดอยู่ (เสียงเพลงที่ Terminal ยังดังอยู่) ก็ไม่ต้องกดรหัสผ่าน เนื่องจากต้องการความรวดเร็วในการที่จะผ่านเข้าออก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อมีการรูดบัตรที่ Terminal แล้ว จะต้องทำการเลือกว่าเป็นการเข้าทำงาน หรือเป็นการบันทึกเวลาออกจากงาน และกดรหัส (ถ้าต้องกด) แล้ว จากนั้น Terminal จะทำการส่งข้อมูลดังกล่าวนี้ไปให้ Center เพื่อให้ตรวจสอบการเข้าออก และ Center จะส่งผลกลับมาที่ Terminal ว่าสามารถผ่านได้หรือไม่ได้ และแสดงผลที่ได้ออกที่หน้าจอแสดงผลของ Terminal ถ้าสามารถผ่านได้ ก็จะส่งเอาท์พุทไปที่ Solinoid ที่ใช้ในการเปิดประตูให้ทำการปลดตัวล็อคประตู ให้สามารถผ่านเข้าไปได้ พร้อมกับทำการสร้างเสียงเพลง โดยประตูจะเปิดเป็นเวลาประมาณ 15 วินาที ถ้าหลังจาก 15 วินาที นี้แล้ว ประตูยังไม่ปิด (โดยใช้ไมโครสวิทช์เป็นเซ็นเซอร์ในการตรวจสอบประตูปิด) Terminal ก็จะสร้างสัญญาณเตือนออกมาเพื่อให้ทำการปิดประตู

ในการบันทึกเวลาเข้าทำงาน จะบันทึกเวลาเข้าทำงานเมื่อมีการผ่านเข้าไปครั้งแรก และอยู่ในช่วงเวลาของการบันทึกเวลาเข้าทำงานด้วย ซึ่งในการเข้าครั้งต่อไป จะไม่ทำการบันทึกเวลาลงไปใหม่อีก ส่วนในการบันทึกเวลาออกจากงาน จะบันทึกเวลาลงไปทุกครั้งที่พนักงานต้องการจะบันทึกเวลาออกจากงาน

<u>บัตรที่ใช้ในการเข้าออกนี้ มีอยู่ 2 แบบคือ</u>

- แบบที่มีการบันทึกเวลาเข้าออกในการทำงาน โดยแบบนี้จะใช้กับพนักงานที่จะต้องมีการบันทึกเวลาเข้าออกทำงานไว้ เพื่อนำไปคิดเป็นเงินเดือน

- แบบที่ไม่ต้องมีการบันทึกเวลาเข้าออก (Extra Type) โดยแบบนี้จะใช้กับบุคคลต่าง ๆ หรือพนักงานที่สามารถเข้าไปที่ Terminal ต่าง ๆ ได้เป็นกรณีพิเศษ และไม่ต้องบันทึกเวลา

<u>การทำงานของ Center</u>

เนื่องจาก Center เป็นส่วนที่เก็บข้อมูลของบัตรที่จะใช้ในการเข้าออก และ เวลาเข้า-ออกของพนักงานทั้งหมด ดังนั้น ในส่วนฮาร์ดแวร์ ของ Center ควรจะมีการ เก็บข้อมูลได้มาก ดังนั้นจึงใช้ CPU เบอร์ Z80180 ซึ่งมีการอ้างข้อมูลได้ถึง 1 เม็กกะไบต์ อีกทั้งในตัว CPU ยังมีอุปกรณ์ต่าง ๆ ที่ใช้ในการทำงานหลายตัวด้วย เช่น

- มีช่องการสื่อสารแบบอนุกรม ถึง 2 ช่อง ซึ่งจะไปใช้ในการติดต่อระหว่าง คอมพิวเตอร์ และ Terminal

- มีวงจรนับภายใน 2 ช่อง ซึ่งจะนำมาใช้ในการ ตั้งค่าเวลาต่าง ๆ ได้อย่างถูกต้อง

- มีวงจร DMA (ไม่ได้ใช้ในงานนี้)

- มี การทำงานในแบบ Sleep Mode , Stop Mode โดยจะหยุดการทำงานของ CPU และอุปกรณ์ภายใน CPU เพื่อเป็นการประหยัดพลังงาน ในขณะที่ไม่มีการใช้เครื่อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อุปกรณ์และลักษณะต่าง ๆ ของตัว Center จะมีดังต่อไปนี้

- สามารถใส่ RAM เพื่อเก็บข้อมูลของบัตร ได้ 64 kbyte ซึ่งจะเก็บได้ 4096 ใบ (บัตร 1 ใบ ใช้ข้อมูล 16 ไบต์)

- มี พอร์ตใช้งาน 3 พอร์ต โดยจะใช้เบอร์ 82C55

- มีวงจร RTC (Real Time Clock) เพื่อใช้เป็นฐานเวลาให้ระบบ โดยจะใช้ IC # MSM 6242 B ซึ่งเป็น RTC ขนาด 4 บิท Address และ 4 บิท Data สามารถตั้งให้ เกิดการ Interupt ได้ 1/64 second, 1 second, 1 minute และ 1 hour และยังมีการแสดงวัน,เดือน, ปี และวันในรอบสัปดาห์

- มีวงจร Power on reset ใช้เบอร์ DS1232

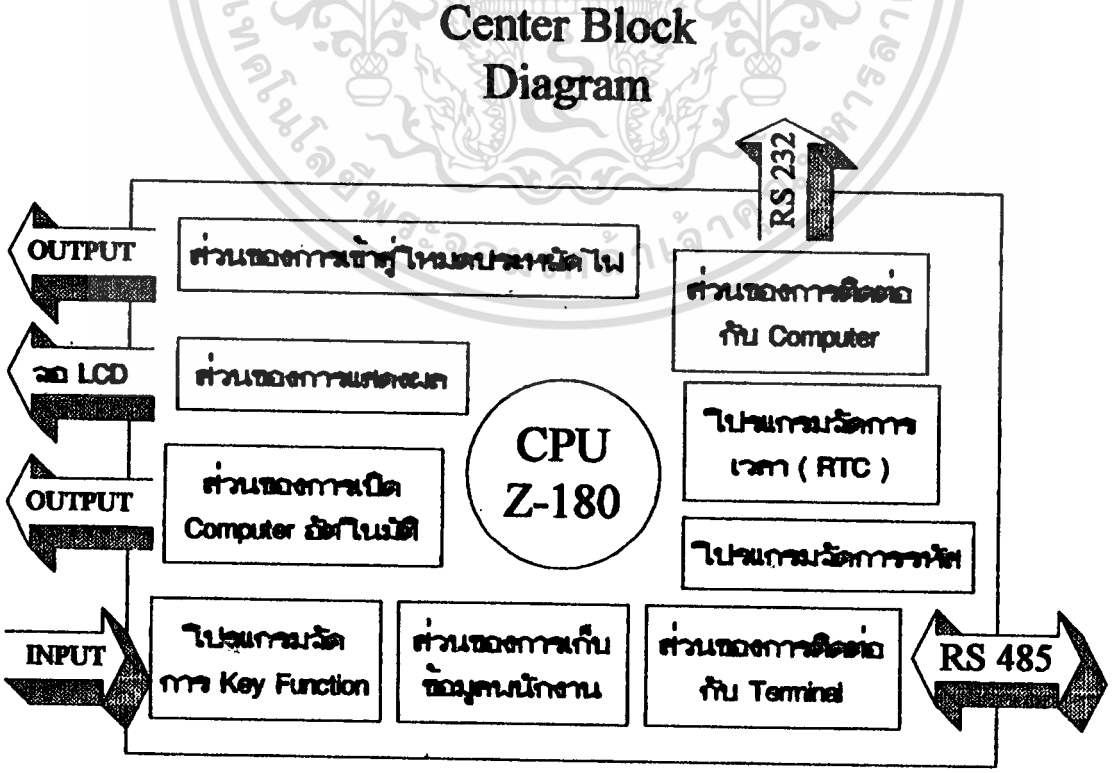
- มี LCD แบบ Dot Matrix ขนาด 16 char จำนวน 2 แถวเป็นตัวแสดงผล

- มี Key ไว้ใช้ในการตั้งค่าต่าง ๆ ของเครื่องโดยประกอบด้วย Key 0-9, Enter, ESC, Del, Function

- มีเครื่องอ่านบัตรแบบอัตโนมัติ พร้อมด้วยวงจรควบคุมและวงจรสร้างสัญญาณถอดรหัสของบัตร ซึ่งมีไว้ใช้ในการอ่านบัตรใหม่ เพื่อจะใช้เป็นบัตรที่ใช้ในการเข้าออก โดยจะทำการบันทึกข้อมูลลงไปใน memory

- มี Battery ที่ใช้ในการ Back up ข้อมูลของ memory และ RTC

โดยโครงสร้างของ Center แสดงได้ดังรูปที่ 4.1



รูปที่ 4.1 โครงสร้างของ Center

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การเก็บข้อมูลของบัตรที่จะผ่านเข้าออก

ในการเก็บข้อมูลของบัตรที่จะผ่านเข้าออก จะทำการเก็บลงใน memory ของ Center โดยบัตร 1 ใบจะใช้เนื้อที่ในการเก็บ 16 ไบต์ ซึ่งจะแบ่งประเภทของบัตรเป็น 2 ชนิดคือ

1. แบบบันทึกเวลาเข้าออก แบบนี้จะต้องมีการบันทึกเวลาเข้าออกในการทำงาน และจะเข้าได้ใน Terminal ที่กำหนดเพียงตัวเดียว โดยมีรูปแบบในการเก็บข้อมูลดังนี้

Card Data (8 byte)	Pass Code (2 byte)	Time work in (2 byte)	Time work out (2 byte)	Personel Code (2 byte)

2. แบบไม่บันทึกเวลาเข้าออก (Extra Type) แบบนี้สามารถเข้าไปที่ Terminal ตัวใดก็ได้ขึ้นอยู่กับการตั้งค่าบัตรในครั้งแรก โดยมีรูปแบบในการเก็บข้อมูลดังนี้

Card Data (8 byte)	Pass Code (2 byte)	Pass Terminal (2 byte)	Personel Code (2 byte)

โดยรายละเอียดจะมีดังนี้

- Card Data เป็นข้อมูลที่อยู่ในบัตร โดยจะนำข้อมูลของบัตรมาใส่ไว้จำนวน 8 ไบต์ เพื่อใช้ในการเปรียบเทียบบัตรที่จะมีการเข้าออก

- Pass Code เป็นรหัสที่ใช้ในการผ่านเข้าออก โดยเป็นตัวเลขฐาน 10 จำนวน 4 ตัว

- Personel Code เป็นรหัสที่ใช้ประจำบัตร และจะเป็นรหัสที่จะใช้ประจำตัวของพนักงานในส่วนของการคิดเงินเดือนในคอมพิวเตอร์ด้วย และยังใช้รหัสนี้ในการอ้างถึงเมื่อจะทำการลบบัตรออกจาก Memory ของ Center ด้วย โดยจะเป็นตัวเลขฐาน 10 จำนวน 4 ตัว

- Time in,out เป็นส่วนที่ใช้เก็บข้อมูลเวลาเข้าออก ของบัตรแบบบันทึกเวลาเข้าออก โดย 2 ไบต์แรกจะเก็บค่าเวลาเข้าทำงาน ส่วน 2 ไบต์หลังจะเก็บค่าเวลาออกจากงาน

- Pass Terminal เป็นส่วนที่แสดงตำแหน่ง Terminal ที่บัตรใบนั้นสามารถจะเข้าได้ โดย ถ้าค่าบิทในข้อมูล 2 ไบต์นี้ จะหมายถึงการเข้าได้ หรือไม่ได้ของ Terminal ตัวที่บิทนั้นอ้างถึงเช่น บิท 7 ของไบต์ 1 จะอ้างถึง Terminal #16 โดยถ้าบิทมีค่าเป็น "1" ก็หมายความว่าสามารถเข้าใน Terminal นั้นได้ แต่ถ้าเป็นบิท "0" ก็แสดงว่าไม่สามารถเข้าใน Terminal นั้นได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถ้าข้อมูลของบัตรในส่วนของ Card Data นี้มีค่าเป็น FFH ก็จะเป็นการบอกว่า memory ที่เก็บข้อมูลของบัตรที่ตำแหน่งนั้นว่างอยู่ สามารถใส่บัตรเพิ่มเข้าไปได้ ถ้าข้อมูลของบัตรนี้มีค่าเป็น EFH ก็แสดงว่าไม่สามารถใส่บัตรเพิ่มเข้าไปได้อีกแล้ว (บัตรเต็ม)

การเก็บค่าสถานะของเครื่อง (Config)

ค่าสถานะของเครื่องคือค่าที่ตั้งให้กับระบบ โดยจะแบ่งเป็น ส่วนของ Center และส่วนของ Terminal ซึ่งจะเก็บอยู่ใน memory โดยมีรายละเอียดดังนี้

1. ค่าสถานะของ Center ประกอบไปด้วย
   - จำนวน Terminal ทั้งหมดที่ต่ออยู่ในระบบ
   - เวลาเปิดเครื่อง Center
   - เวลาปิดเครื่อง Center
   - เวลาส่งข้อมูลให้คอมพิวเตอร์
2. ค่าสถานะของ Terminal จะแยกเป็น Terminal แต่ละตัว ซึ่งประกอบไปด้วย
   - Mode การใช้งานของ Terminal ตัวนั้น
   - ช่วงเวลาการบันทึกเวลาเข้าทำงาน
   - ช่วงเวลาการบันทึกเวลาออกจากงาน
   - Address เริ่มต้นของ ข้อมูลของบัตรประจำ Terminal ตัวนั้น

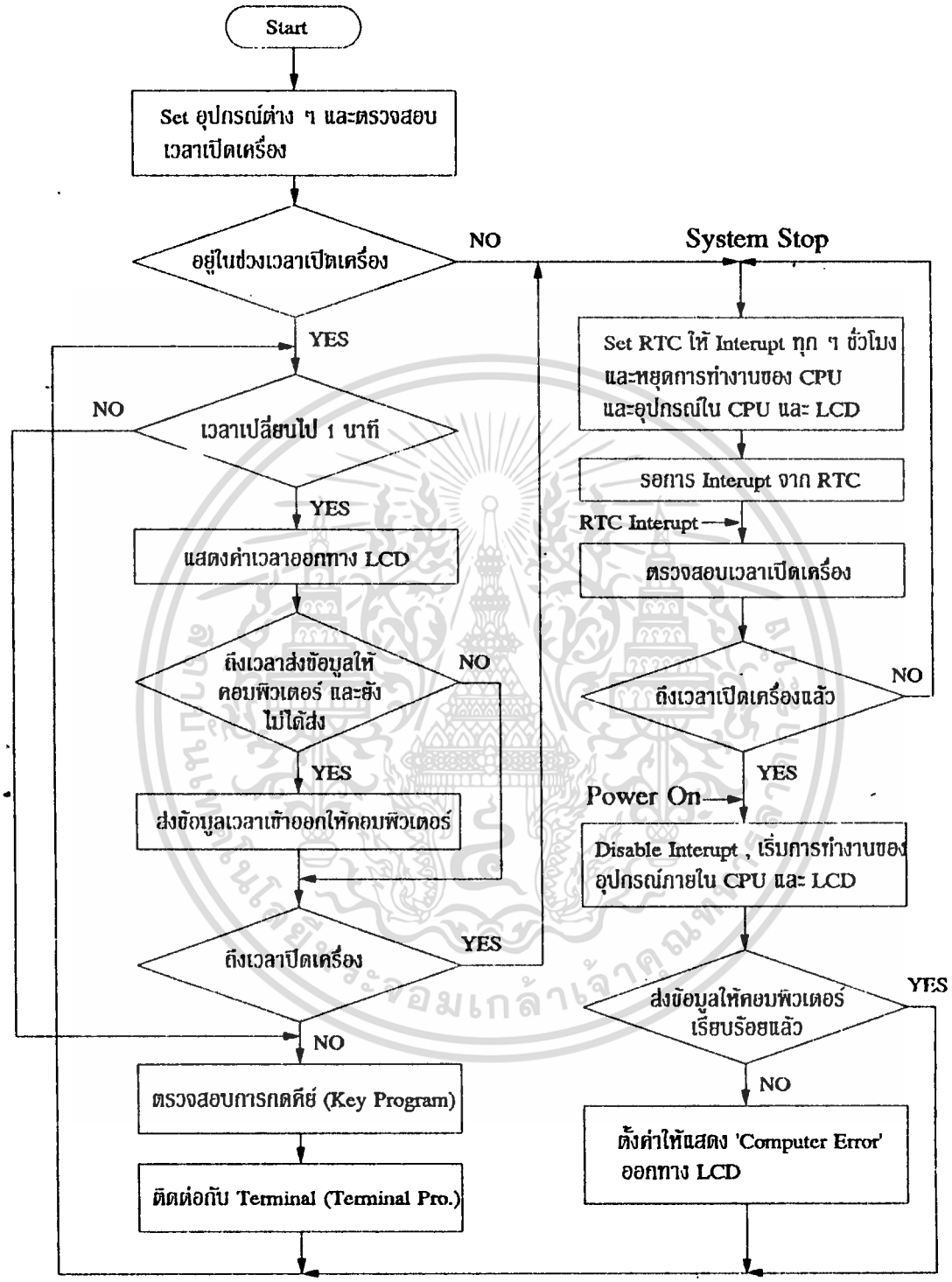
หน้าที่ต่าง ๆ ของ Center ประกอบด้วย

- ติดต่อกับ Terminal เพื่อตรวจสอบและควบคุมการเข้าออก
- บันทึกเวลาเข้าออก ของพนักงาน
- ติดต่อกับคอมพิวเตอร์ เพื่อส่งค่าเวลาเข้าออกของพนักงานให้คอมพิวเตอร์
- ติดต่อกับ Key SW เพื่อแก้ไขค่าต่าง ๆ ในตัว Center
- ตรวจสอบเวลาต่าง ๆ เช่น เวลาเปิด-ปิดเครื่อง, เวลาส่งข้อมูลให้คอมพิวเตอร์ เป็นต้น

โดยในส่วนของโปรแกรมการทำงานของ Center จะแสดงได้ดัง Flow Chart ดังรูปที่ 4.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Center Program



รูปที่ 4.2 FlowChart แสดงโปรแกรมการทำงานของ Center

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานของโปรแกรม ของ Centerจะแบ่งเป็นส่วน ๆ คือ

1. การทำงานเมื่อเกิดการ Reset

เมื่อเกิดการ Reset โปรแกรมจะทำการเซ็ทอุปกรณ์ทั้งหมดให้ทำงานตามที่กำหนดไว้ เช่น การทำงานของ CPU, Port serial, Port in-out, Counter, RTC จากนั้นก็จะตรวจสอบเวลาว่าเวลาในขณะนั้น อยู่ในช่วงเวลาของการเปิดเครื่องหรือไม่ ถ้าอยู่ในช่วงเวลาก็จะเข้าสู่การทำงานของโปรแกรม แต่ถ้ายังไม่อยู่ในช่วงเวลาก็จะเข้าสู่โหมดของ System stop

2. การทำงานในโหมด System stop

เป็นโหมดที่เครื่องหยุดทำงานในช่วงเวลาที่ไม่มีการใช้งาน (ช่วงเวลาปิดเครื่อง) โดยในโหมดนี้ ตัว CPU จะอยู่ในคำสั่ง Sleep mode (มีเฉพาะใน CPU เบอร์ Z80180) และอุปกรณ์ในตัว CPU เช่น Port Serial, counter เป็นต้น จะอยู่ในสภาวะ Stop Mode คือหยุดการทำงานของอุปกรณ์ใน CPU ทั้งหมด และยังทำการปิดหน้าจอแสดงผล LCD ด้วย ทั้งนี้เพื่อต้องการประหยัดพลังงานของเครื่อง และจะตั้งค่า RTC ให้ทำการ Interupt ทุก ๆ 1 ชั่วโมง ซึ่งก็คือ CPU จะออกจากสภาวะนี้เมื่อมีการ Interupt จาก RTC เมื่อเกิดการ Interupt โปรแกรมก็จะไปตรวจสอบการส่งข้อมูลให้คอมพิวเตอร์ ถ้ายังไม่ได้ส่งก็จะทำการส่งข้อมูลให้คอมพิวเตอร์ (การส่งข้อมูลให้คอมพิวเตอร์ จะอธิบายในส่วนของบทที่ 5) จากนั้นก็จะไปตรวจสอบเวลาที่เปิดเครื่อง ถ้ายังไม่ถึงเวลาเปิดเครื่องก็จะกลับเข้าสู่สภาวะ Stop mode อีก แต่ถ้าถึงเวลาเปิดเครื่องแล้ว ก็จะเข้าสู่โปรแกรมส่วนของ Power on ดังแสดงใน Flow Chart

3. การทำงานส่วนของ Power on

การทำงานในส่วนนี้เกิดการ Interupt จาก RTC และโปรแกรมตรวจพบว่าถึงเวลาที่จะต้องเปิดเครื่องแล้ว โดยโปรแกรมส่วนนี้จะทำการตั้งค่าให้อุปกรณ์ใน CPU ออกจาก Stop Mode และเปิดหน้าจอแสดงผล LCD และเซ็ท RTC ไม่ให้มีการ Interupt จากนั้นก็จะตรวจสอบว่าส่งข้อมูลให้กับคอมพิวเตอร์แล้วหรือยัง ถ้ายังไม่ได้ส่งก็แสดงว่าระบบการส่งข้อมูลให้คอมพิวเตอร์เกิดการผิดพลาดไม่สามารถส่งข้อมูลได้ และจะทำการแสดงผลที่ LCD ว่าเกิดการผิดพลาดในการส่งข้อมูล และจะทำการลบค่าเวลาเข้าออกทำงานของของเก่าออกด้วย

4. การทำงานส่วนของการตรวจสอบเวลา

การทำงานในส่วนนี้ จะทำการตรวจสอบเวลาต่าง ๆ โดยจะทำการตรวจสอบเมื่อเวลาจาก RTC เปลี่ยนไปทุก ๆ 1 นาที โดยจะอ่านค่าเวลามาจาก RTC และนำค่าเวลาที่อ่านได้นี้ไปทำการเปรียบเทียบกับค่าเวลาต่าง ๆ ที่จะต้องทำงาน ถ้าตรงกัน ก็จะเข้าไปทำงานที่โปรแกรมนั้น ๆ โดยค่าเวลาต่าง ๆ ที่จะต้องมีการตรวจสอบการทำงาน คือ

- ตรวจสอบเวลาที่จะต้องทำการส่งข้อมูลให้คอมพิวเตอร์
- ตรวจสอบเวลาที่จะต้องปิดเครื่อง คือเมื่อถึงเวลาปิดเครื่องก็จะเข้าสู่ในโปรแกรมส่วนของ system stop
- จะมีการแสดงผลค่าเวลา ออกทาง LCD Display

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. การติดต่อกับผู้ใช้เครื่อง ผ่านทาง Key SW

เป็นโปรแกรมส่วนที่ใช้ในการแก้ไขค่าต่าง ๆ ของ Center โดยผ่านทาง Key SW โดยจะใช้ คีย์ จำนวน 16 คีย์ ในการติดต่อกับผู้ใช้

โดย Key ที่ใช้จะประกอบด้วย

- Key number 0-9 เป็นคีย์ที่ใช้ป้อนค่าต่าง ๆ
- Key Enter เป็นคีย์ที่ใช้เมื่อมีการตกลง หรือเมื่อจะใส่ค่าต่าง ๆ เข้าไป
- Key ESC เป็นคีย์ที่ใช้ในการยกเลิก หรือออกจาก Function แก้ไข
- Key Function เป็นคีย์ที่ใช้ในการเข้าสู่ Function ต่าง ๆ
- Key Del เป็นคีย์ที่ใช้ลบค่าที่ป้อนเข้าไปทีละตัว ในกรณีที่ป้อนค่าผิด
- Key Yes เป็นคีย์ที่ใช้ในการตอบตกลงเมื่อมีการถาม
- Key No เป็นคีย์ที่ใช้ในการไม่ตกลงเมื่อมีการถาม
- Key Left,Right เป็นคีย์ที่ใช้ในการเลือกค่าต่าง ๆ

โดย Key Yes-Left และ No-Right จะอยู่บนคีย์เดียวกัน

โดยโปรแกรมในส่วนนี้จะทำการตรวจสอบการกดคีย์โดยใช้วิธีการ Scan การกดคีย์แบบ Output 4 แถว และ Input 4 แถว ซึ่งในส่วนของโปรแกรมจะต้องมีส่วนตรวจสอบการปล่อยคีย์, การลดเบาน์ซ์ที่เกิดจากการกดคีย์ เป็นต้น เพื่อให้การกดคีย์เป็นไปอย่างถูกต้อง และจะมีการแสดงผลการกดคีย์ทางหน้าจอแสดงผล LCD เพื่อติดต่อกับผู้กดคีย์ด้วย

โดยในการใช้งานจะแบ่งเป็น Function จำนวน 4 Function ดังนี้

5.1 Function 1 ใช้ในการแก้ไขค่าต่าง ๆ ของ Center และ Terminal ดังนี้

5.1.1 แก้ไขค่าของ Center

- แก้ไขเวลาเปิดเครื่อง Center
- แก้ไขเวลาปิดเครื่อง Center
- แก้ไขเวลาส่งข้อมูลให้คอมพิวเตอร์

5.1.2 แก้ไขค่าของ Terminal

ในการแก้ไขค่าของ Terminal จะแบ่งการแก้ไขเป็นตัว ๆ ไปคือ

- แก้ไข Mode การใช้งานของ Terminal
- แก้ไขช่วงเวลาที่กำหนดให้มีการบันทึกเวลาเข้าทำงาน
- แก้ไขช่วงเวลาที่กำหนดให้มีการบันทึกเวลาออกทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2 Function 2 ใช้ในการเพิ่มบัตร และแก้ไขรหัสของบัตรที่มีอยู่แล้วที่ใช้ในการเข้าออก คือจะมีการตรวจสอบบัตรที่จะทำการเพิ่มหรือจะทำการแก้ไข กับข้อมูลของบัตรที่มีอยู่ใน Center ว่าเป็นบัตรที่มีข้อมูลอยู่ใน memory ของ Center หรือไม่ ถ้าเป็นบัตรที่อยู่ใน Center ก็แสดงว่าเป็นการเปลี่ยนแปลงหรือแก้ไขค่ารหัสของบัตร แต่ถ้าเป็นบัตรใหม่ก็จะทำการเพิ่มบัตรเข้าไป

5.3 Function 3 ใช้ในการลบบัตรที่มีอยู่ใน memory ของ Center โดยการทำงานของ Function นี้คือจะนำค่ารหัสประจำตัวของบัตรที่ป้อนผ่าน Key SW มาทำการเปรียบเทียบกับรหัสประจำตัวของบัตร โดยจะเปรียบเทียบกับข้อมูลของบัตรแบบ Extra Type ก่อน ถ้ามีรหัสตรงกัน ก็จะแสดงข้อความถามว่า ต้องการลบบัตรใบนี้หรือไม่ ถ้าตกลงก็จะทำการลบบัตรนี้ออกจาก memory และทำการเลื่อนข้อมูลของบัตรใบที่อยู่ข้างหลังเข้ามาแทนที่ จากนั้นจึงไปเปรียบเทียบกับข้อมูลของบัตรแบบที่มีการตรวจสอบเวลาเข้าออก โดยเริ่มจาก memory ของ Terminal ตัวแรกสุดก่อน ถ้ามีข้อมูลบัตรตรงกับข้อมูลของบัตรใน memory ของ Terminal ตัวใด ก็จะแสดงผลไปถามว่า ต้องการลบหรือไม่ ถ้าต้องการลบก็ลบข้อมูลบัตรใบนั้นออกและเลื่อนข้อมูลของบัตรที่อยู่ข้างหลังขึ้นมาเช่นกัน

5.4 Function 4 ใช้ในการตั้งเวลาของ RTC ให้ตรงกับเวลาจริง

6. การติดต่อกับ Terminal

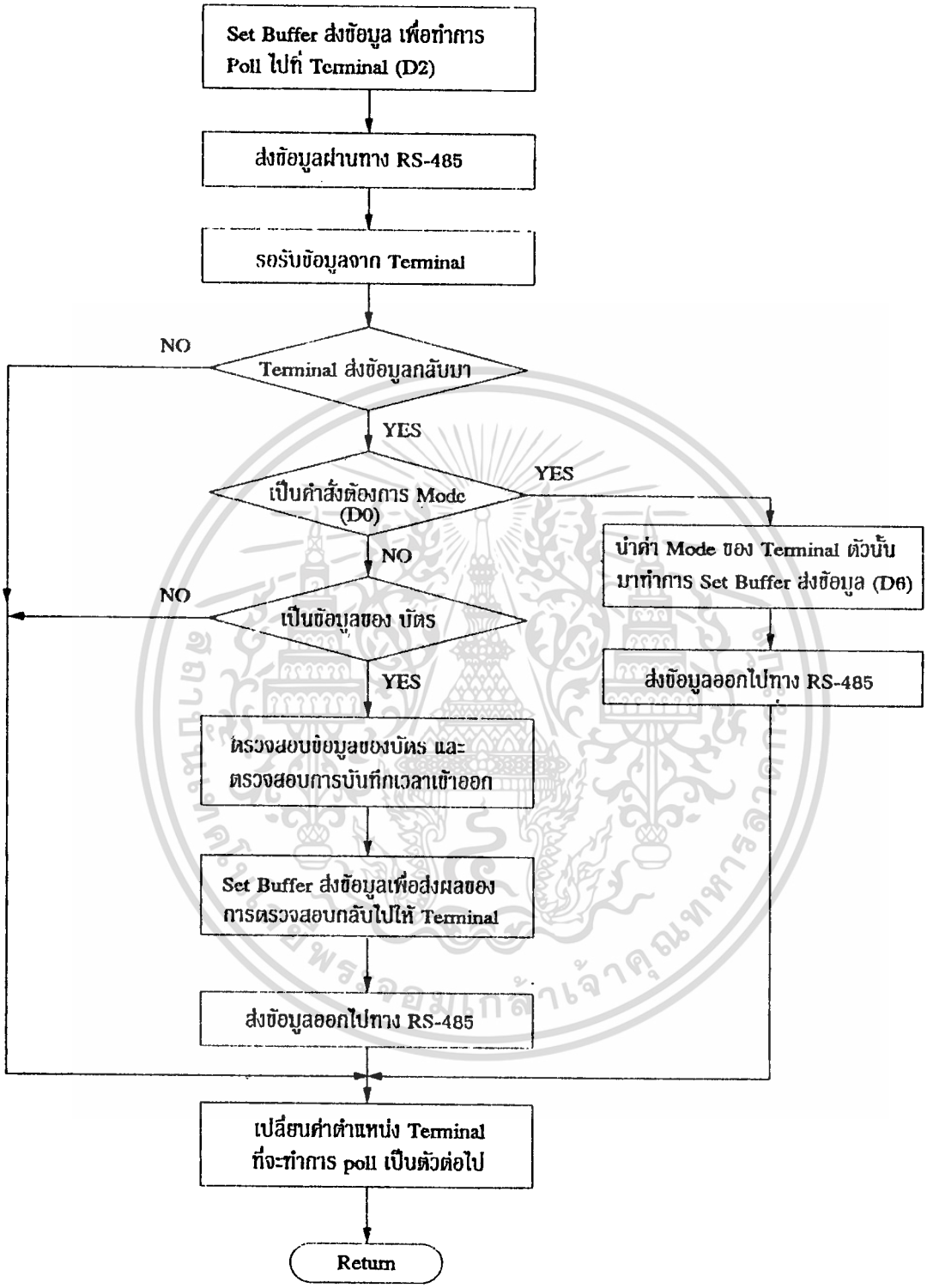
จะมีการติดต่อกับ Terminal ผ่านทางมาตรฐาน RS-485 ซึ่งวิธีการติดต่อนี้ได้แสดงไว้ในบทที่ 3 โปรแกรมส่วนของการติดต่อกับ Terminal แสดงได้ดัง Flow Chart ในรูปที่ 4.3

จาก Flow Chart ในรูปที่ 4.3 โปรแกรมจะทำการ Poll ไปที่ Terminal โดยการใช้คำสั่ง D2H (แสดงไว้ในบทที่ 3) เมื่อส่งข้อมูลออกไปแล้ว ก็จะรอการตอบรับจาก Terminal ตัวนั้นภายในเวลาที่กำหนด ถ้าไม่มีการตอบรับกลับมาก็จะทำการเปลี่ยนตัวค่าของ Terminal ที่จะทำการติดต่อ ใน Byte Terminal และออกจากโปรแกรม แต่ถ้ามีการตอบรับกลับมา ก็จะตรวจสอบว่าข้อมูลที่ Terminal ส่งกลับมาเป็นคำสั่งอะไร ซึ่งคำสั่งที่ Center จะต้องทำการตอบรับและส่งค่าตอบรับกลับไปให้ Terminal จะประกอบไปด้วย

1. คำสั่งต้องการ Mode (D1H) Center จะต้องนำค่า Mode ที่อยู่ในค่าสถานะ (config) ของ Terminal ตัวนั้น มาทำการเซ็ท Buffer ส่ง ในคำสั่ง ส่ง Mode ไปให้Terminal (D6H) (คำสั่งนี้แสดงไว้ในบทที่ 3)

2. คำสั่งให้ตรวจสอบการเข้าออกจากข้อมูลบัตรที่ส่งมา (D3H) Center จะต้องนำข้อมูลของบัตรที่ส่งมาจาก Terminal มาเปรียบเทียบกับข้อมูลของบัตรที่อยู่ใน memory ที่เก็บข้อมูลบัตรของ Terminal นั้น ถ้าข้อมูลของบัตรที่ส่งมามีข้อมูลอยู่ใน memory ก็จะทำการตรวจสอบรหัสผ่าน (เมื่อมีการกดรหัส) ถ้าถูกต้อง ก็จะทำการตรวจสอบว่าจะทำการเข้าหรือออก (in,out) โดยตรวจสอบเวลาว่าในเวลานี้ อยู่ในช่วงเวลาของการบันทึกเวลาเข้าหรือออก หรือไม่ (โดยดูจากค่าสถานะของ Terminal ตัวนั้น) ถ้าอยู่ในช่วงเวลา ก็จะทำการบันทึกเวลาเข้าหรือออก ให้ แต่ถ้าไม่อยู่ในช่วงเวลานี้ ก็จะไม่บันทึกเวลาให้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.3 Flow Chart แสดงโปรแกรมการติดต่อกับ Terminal

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และจะแจ้งผลกลับไปให้ Terminal ตัวนั้น โดยใช้คำสั่ง D8H (คำสั่งนี้แสดงไว้ในบทที่ 3)

ถ้าข้อมูลของบัตรไม่มีอยู่ใน memory ของ Terminal นั้น โปรแกรมก็จะไปตรวจสอบกับข้อมูลของบัตรใน memory ของบัตรแบบ Extra Type ถ้ามีข้อมูลของบัตรตรง และรหัสตรง (เมื่อมีการกดรหัส) ก็จะทำการตรวจสอบว่าบัตรใบนี้สามารถผ่านเข้าใน Terminal ตัวนี้ได้หรือไม่ และจะส่งผลกลับไปให้ Terminal ตัวนั้น

7. การติดต่อกับคอมพิวเตอร์

ในการติดต่อกับคอมพิวเตอร์ เกิดขึ้นเมื่อ Center ต้องการส่งค่าเวลาเข้าออกทำงานของพนักงานไปให้คอมพิวเตอร์ โดยจะทำการส่งข้อมูลทุกวัน ซึ่งจะทำการส่งข้อมูลเมื่อสิ้นสุดการทำงานในแต่ละวัน โดยจะกำหนดเวลาส่งจากค่าสถานะของ Center

รูปแบบของข้อมูลที่ใช้ในการส่งข้อมูลให้คอมพิวเตอร์มีดังนี้

STX	Data 1	Data 2 ..	..Data n	ETX	CRC	ETB,EOT

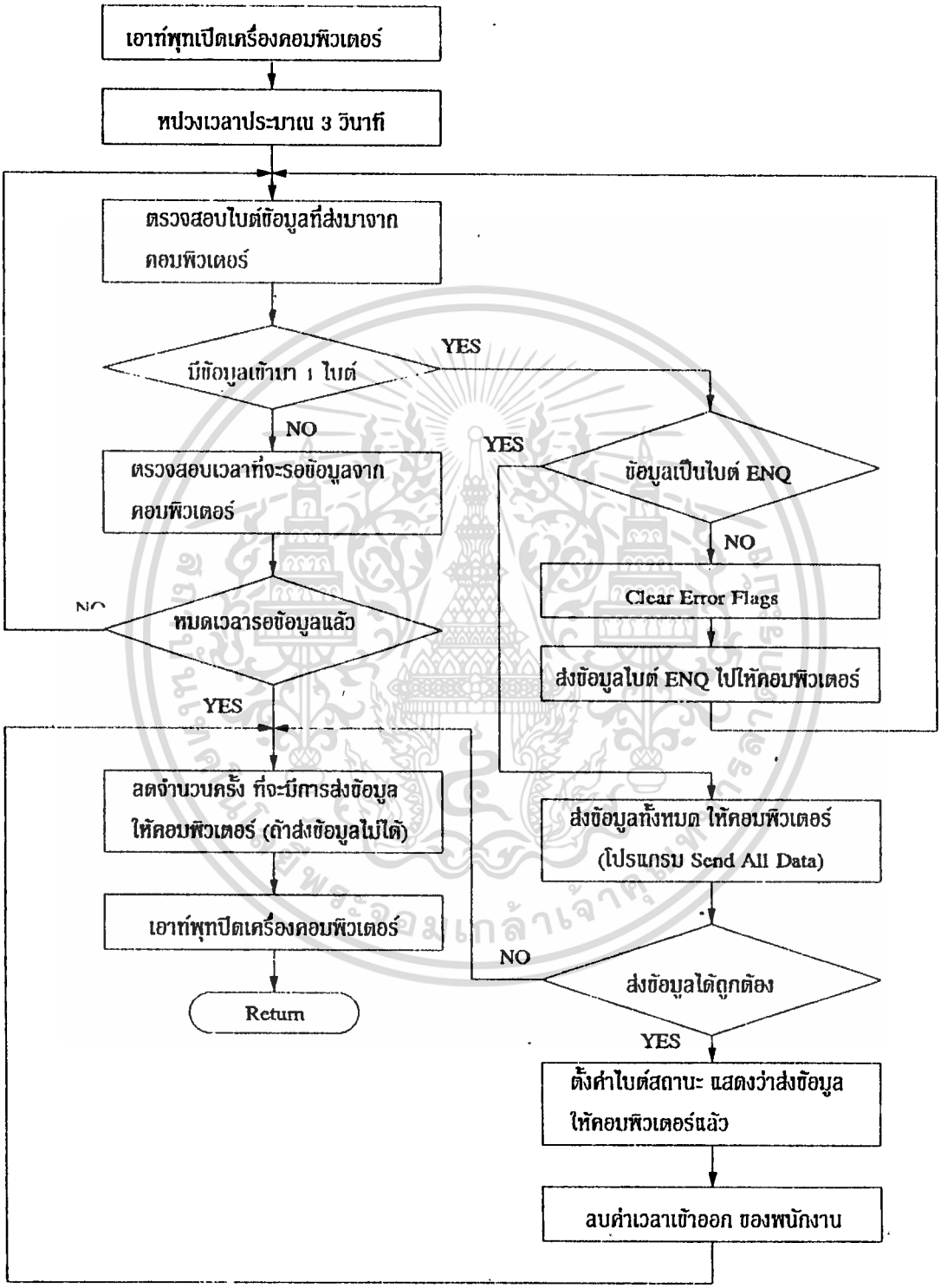
ในส่วนของ Data จะเป็นข้อมูลเวลาเข้าออกของพนักงาน 1 คนโดยจะประกอบไปด้วยข้อมูลดังต่อไปนี้

Personel Code	Day in week	Day	Time work in	Time work out

เนื่องจากข้อมูลที่จะส่งให้คอมพิวเตอร์มีจำนวนมาก ดังนั้นในการส่งข้อมูลให้คอมพิวเตอร์นี้จึงแบ่งข้อมูลที่จะส่งออกเป็นบล็อก ๆ โดยบล็อกหนึ่ง ๆ จะมีข้อมูลยาวไม่เกิน 256 ไบต์ และจะมีการตรวจสอบความถูกต้องของข้อมูลในทุก ๆ บล็อก ในการส่งข้อมูลให้คอมพิวเตอร์นี้ แสดงได้จาก Flow Chart ในรูปที่ 4.4

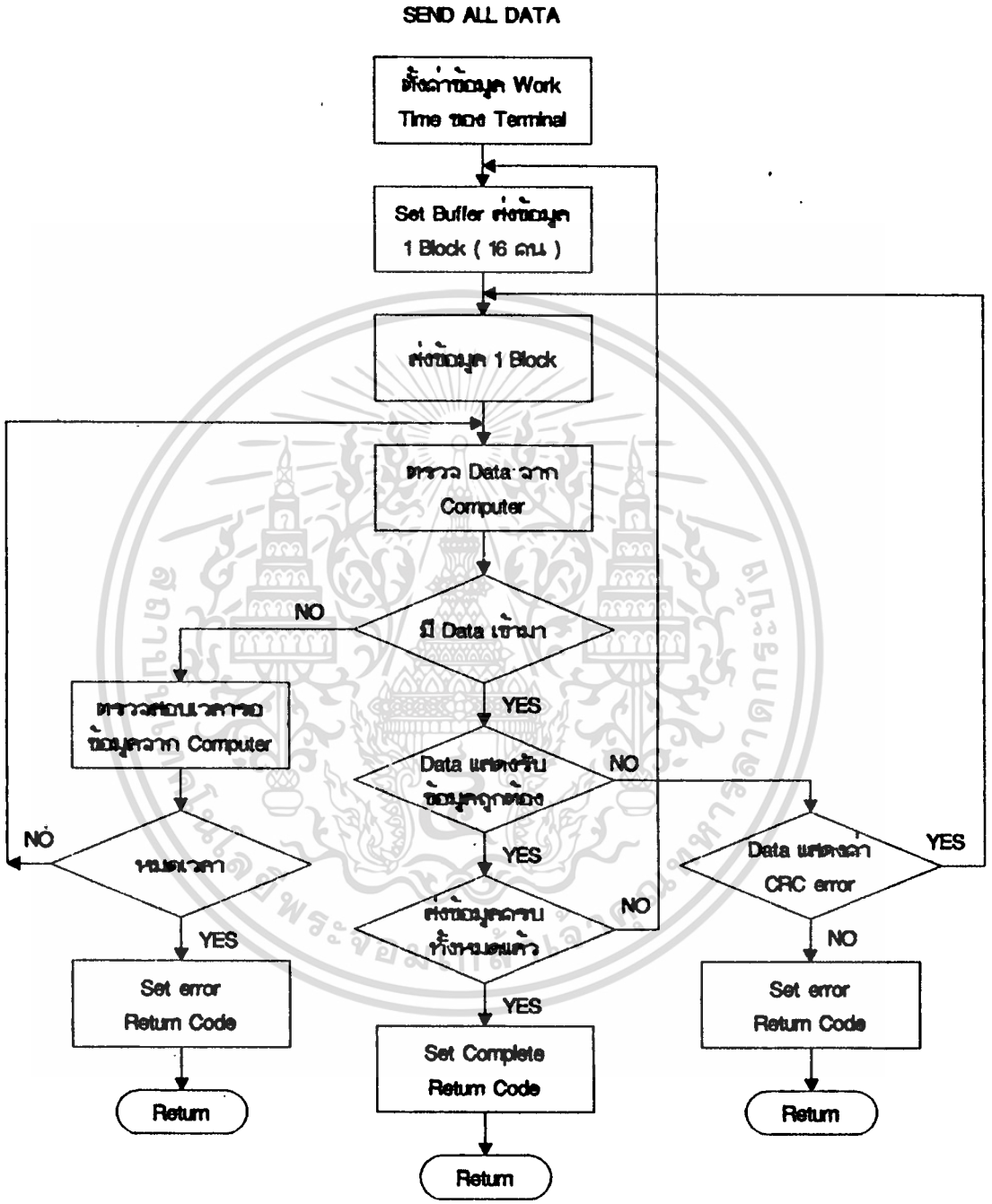
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Send Data To Computer



รูปที่ 4.4 Flow Chart แสดงโปรแกรม Send Data To Computer

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.4 (ต่อ)

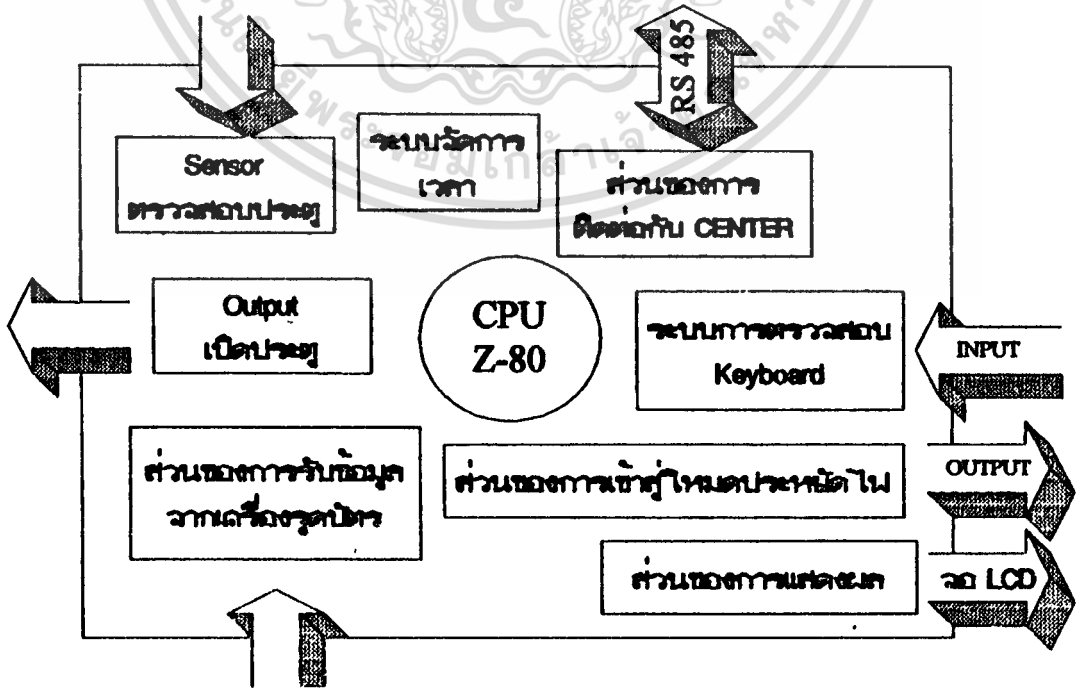
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การส่งข้อมูลให้คอมพิวเตอร์ เริ่มจากส่งค่าเอาท์พุทไปขับรีเลย์ เพื่อเปิดเครื่องคอมพิวเตอร์ โดยในส่วนของตัวคอมพิวเตอร์เมื่อเปิดเครื่องแล้ว จะเข้าไปรันโปรแกรม Autoexec.bat โดยจะให้ไปเข้าสู่โปรแกรมรับข้อมูลจาก Center จากนั้น Center จะทำการรอรับข้อมูลจากคอมพิวเตอร์จำนวน 1 ไบต์ ซึ่งจะกำหนดให้เป็น ENQ (05H) ถ้าไม่ใช่ ENQ Center ก็จะทำการส่งไบต์ ENQ ไปให้คอมพิวเตอร์ เพื่อให้คอมพิวเตอร์ส่งไบต์ ENQ มาใหม่ เมื่อ Center รับไบต์ ENQ จากคอมพิวเตอร์ได้แล้ว ก็จะทำการส่งข้อมูลออกไป 1 บล็อก และจะรอรับข้อมูลจากคอมพิวเตอร์ 1 ไบต์ เพื่อบอกว่าข้อมูลที่รับได้นั้นมีค่าที่ถูกต้องหรือไม่ ถ้าไม่ถูกต้อง Center ก็จะทำการส่งข้อมูลบล็อกนั้นไปให้คอมพิวเตอร์ใหม่ แต่ถ้าถูกต้อง Center ก็จะทำการส่งข้อมูลบล็อกต่อไปให้คอมพิวเตอร์ จนกว่าจะทำการส่งข้อมูลไปจนหมด ถ้าการส่งข้อมูลเป็นไปอย่างถูกต้องทั้งหมด Center จะทำการลบค่าเวลาเข้าออก ทำงานของพนักงาน และทำการส่งเอาท์พุทไปปิดเครื่องคอมพิวเตอร์

ในกรณีที่ไม่มีข้อมูลติดต่อมาจากคอมพิวเตอร์ Center จะมีการกำหนดเวลาที่จะรอข้อมูลจากคอมพิวเตอร์ ประมาณ 2 นาที ถ้าหลังจาก 2 นาทีนี้แล้ว ยังไม่มีการติดต่อมาจากคอมพิวเตอร์ ก็จะถือว่าระบบการส่งข้อมูลผิดพลาด และ Center จะพยายามทำการส่งข้อมูลให้คอมพิวเตอร์อีก ในช่วงเวลาที่อยู่ในสภาวะ Stop Mode

การทำงานของ Terminal

โครงสร้างและอุปกรณ์ต่าง ๆ ของตัว Terminal แสดงได้ดังรูปที่ 4.5



รูปที่ 4.5 โครงสร้างของ Terminal

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 4.5 Terminal จะใช้ CPU เบอร์ Z80 เป็นตัวทำงานของระบบ โดยมีอุปกรณ์และส่วนประกอบที่ใช้ดังนี้

- อุปกรณ์ในการติดต่อกับ Center ซึ่งก็คือการติดต่อทางมาตรฐาน RS-485 โดยจะใช้ IC #8251 ในการสร้างสัญญาณข้อมูลอนุกรม และ IC #75176 ในการสร้างสัญญาณผ่านทาง RS-485

- วงจรสร้างเสียงเพลง และสัญญาณเตือน จะใช้ IC #7910 ในการสร้างเสียงเพลงและสัญญาณเตือน โดยจะมี IC # 4066 ซึ่งเป็นอีเล็คโทรนิคส์สวิทช์ ไปควบคุมในการสร้างเสียงเพลงและสัญญาณเตือน ซึ่งในการควบคุมนี้จะใช้ Port out จาก CPU มาทำการควบคุม IC # 4066

- วงจรควบคุมการเปิดปิดเครื่อง ในการใช้งานของ Terminal นี้ จะออกแบบให้เมื่อไม่มีการใช้งาน หรือมีการรูดบัตรเพื่อที่จะผ่านเข้าออก เป็นเวลานานติดต่อกันประมาณ 10 นาที ตัวเครื่องก็จะปิด คืออยู่ในสภาวะ Standby และในขณะที่ Center ไม่มีข้อมูลส่งมาทาง RS-485 ตัวเครื่องก็จะปิดเช่นกัน ทั้งนี้เพื่อเป็นการประหยัดพลังงานเมื่อไม่มีการใช้เครื่อง ซึ่งในส่วนของการควบคุมการเปิดปิดเครื่องนี้จะใช้ IC #4528 ซึ่งเป็น IC Monostable โดยจะนำเอาท์พุทไปควบคุมการเปิด เครื่อง ซึ่งก็คือเมื่อไม่มีสัญญาณจาก RS-485 มาทำการทริกที่ขาทริกของ IC Monostable เป็นเวลาประมาณ 5 วินาที output ของ Monostable จะเป็น "0" ซึ่งจะไปทำการ off transistor ให้ตัดไฟเลี้ยงของวงจรออก ส่วนในการปิดเครื่องจากตัว CPU เอง จะใช้ IC # 4027 ซึ่งเป็น IC JK-FF ในการควบคุม ซึ่งถ้าจะทำการปิดเครื่อง จะส่งเอาท์พุทไปทำการรีเซ็ท JK-FF ซึ่งจะนำเอาท์พุทนี้ไปทำการรีเซ็ท monostable อีกทีหนึ่ง

- Output เปิดประตู เป็นส่วนที่จะไปทำการเปิดประตู โดยใช้เป็นตัวล็อคแบบใช้ไฟฟ้าในการควบคุม

- Sensor ตรวจสอบประตู เป็น Sensor ที่ใช้ตรวจสอบการปิดประตู เพื่อใช้ควบคุมการส่งเอาท์พุทไปทำการเปิดปิดประตู และใช้ในการสร้างสัญญาณเตือน เมื่อประตูเปิดค้างไว้

- การแสดงผล จะใช้ LCD ขนาด 16 ตัวอักษร จำนวน 2 แถว เพื่อให้แสดงผลต่าง ๆ ในการใช้งาน

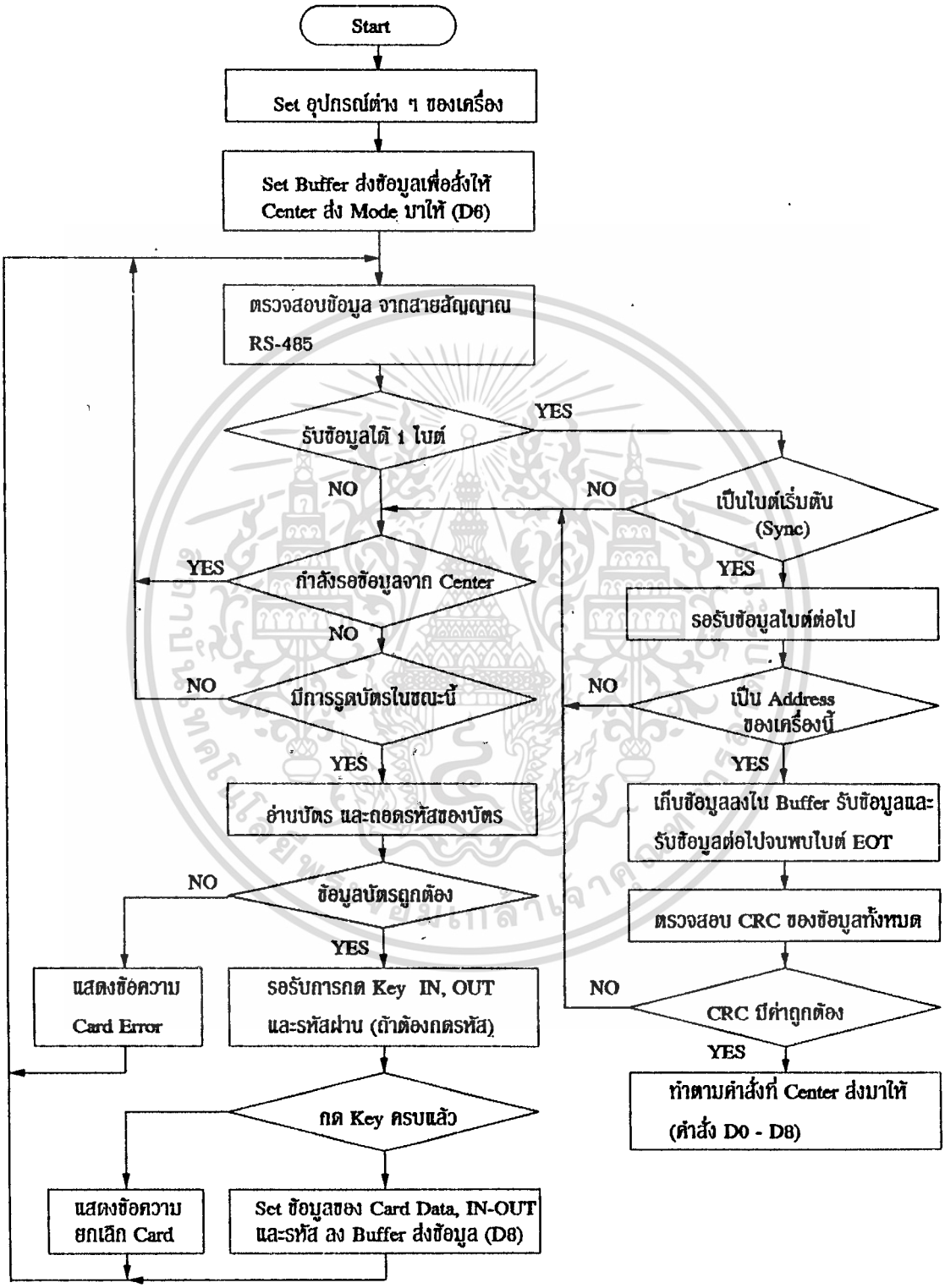
- Key SW เป็น Key ที่ใช้ติดต่อ เมื่อมีการรูดบัตร เช่นใช้ในการกดรหัส เป็นต้น

- ส่วนของเครื่องอ่านบัตร ซึ่งในส่วนนี้จะใช้เครื่องรูดบัตร และจะประกอบไปด้วยวงจรที่สร้างสัญญาณที่ใช้ในการถอดรหัส โดยในส่วนนี้ แสดงไว้ในบทที่ 2

ง. ในส่วนของ Terminal มีหน้าที่คอยตรวจสอบการรูดบัตรเพื่อที่จะผ่านเข้าออก ในบริเวณของ Terminal ตัวนั้น ๆ และควบคุมการเปิด-ปิด ประตู โดยสามารถแสดงการทำงานของโปรแกรมได้จาก Flow Chart ในรูปที่ 4.6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Terminal Program



รูปที่ 4.6 Flow Chart แสดงโปรแกรมของ Terminal

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

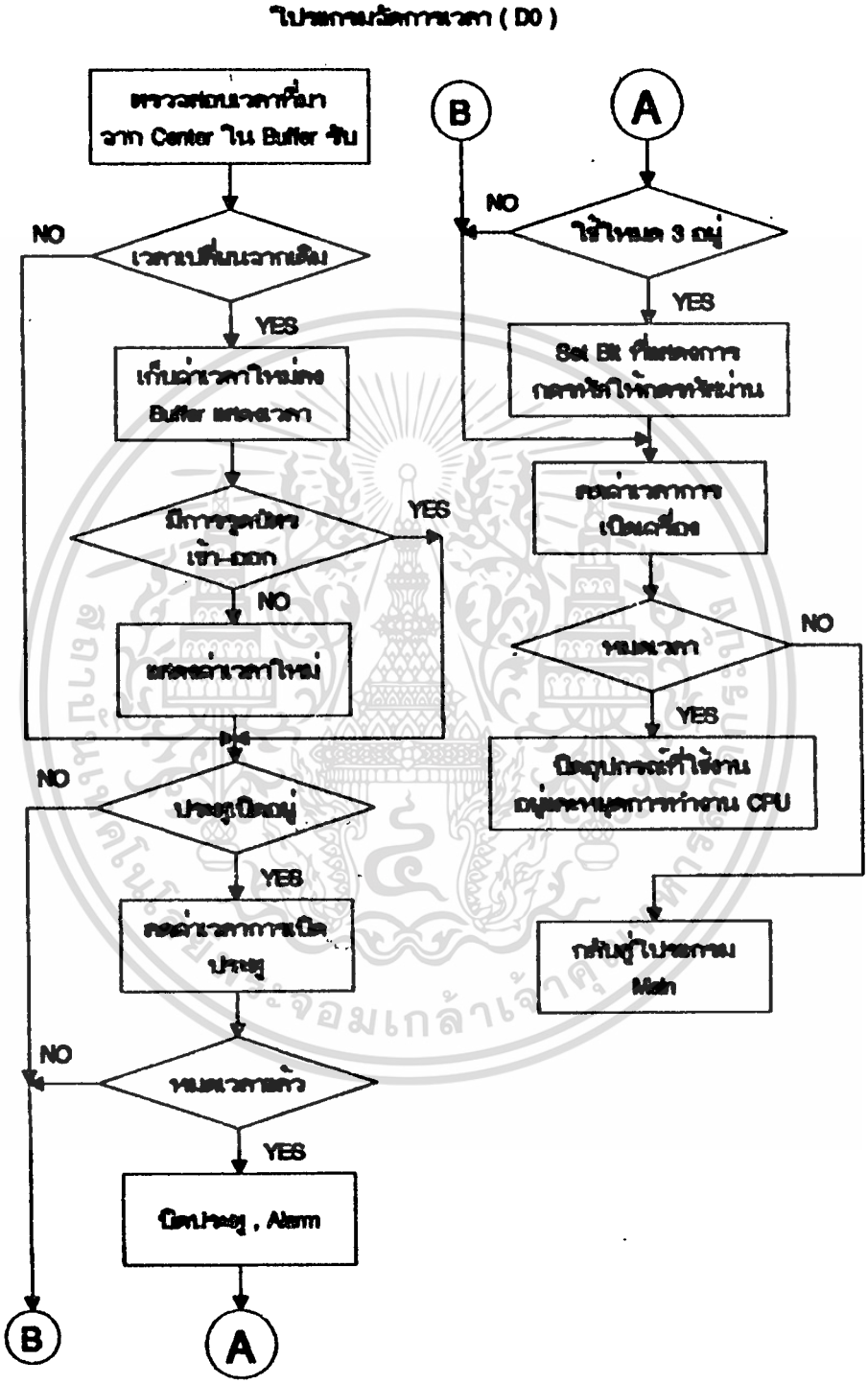
จากรูปที่ 4.6 การทำงานของ Terminal เริ่มจาก เมื่อเปิดเครื่อง หรือรีเซ็ทเครื่องก็จะมีการ set Buffer ส่งข้อมูลไปบอก Center ให้ส่ง Mode มาให้ (คำสั่ง D1H) โดยในส่วนของโปรแกรมหลักจะมีการติดต่อข้อมูลกับ Center กับการติดต่อการอ่านบัตรที่จะผ่านเข้าออก โดยในการติดต่อกับ Center ผ่านทาง RS-485 จะทำการตรวจสอบการ Poll จาก Center เมื่อได้รับการ Poll (ผ่านทางคำสั่ง D2H) แล้ว Terminal จะส่งข้อมูลใน buffer ส่งข้อมูลไปให้ Center ซึ่งจะเป็นคำสั่งต่าง ๆ ดังที่ได้กล่าวในบทที่ 3 โดยในช่วงเวลาที่รอข้อมูลในแต่ละไบต์จาก Center นี้จะต้องมีการตรวจสอบการรูดบัตรด้วยโดยมีการตรวจสอบจากสัญญาณพัลส์ที่เกิดจากการรูดบัตร เนื่องจากในช่วงเวลาที่รอข้อมูลจาก Center อยู่นี้ อาจเป็นช่วงที่มีการรูดบัตรเข้ามาก็ได้ เมื่อทำการรับข้อมูลและโต้ตอบข้อมูลกับ Center ในชุดนั้นเสร็จสิ้นแล้ว ก็จะมาเข้าสู่โปรแกรมการอ่านบัตรทันที เพื่อให้ทันกับข้อมูลของบัตรที่ทำการค่าน เพราะในช่วงแรกของการอ่านบัตร จะเป็นส่วนของสัญญาณ Sync ของบัตร (รายละเอียดอยู่ในบทที่ 2) ซึ่งจะมีความยาวพอสมควร ซึ่งสามารถตัดทิ้งไปได้บ้าง โดยในส่วนของการอ่านบัตรและถอดรหัสบัตร ได้แสดงอยู่ในบทที่ 2 เมื่อทำการอ่านบัตรและถอดรหัสได้ถูกต้องแล้ว ก็จะให้เลือกกดคีย์ว่าจะทำการเข้า หรือออก (ในการเข้าหมายถึงจะทำการบันทึกเวลาเข้าทำงาน หรือการผ่านเข้าไปตามปกติ ส่วนการออก คือการจะบันทึกเวลาออกจากงาน) จากนั้นจะตรวจสอบว่าในขณะนั้นต้องกดรหัสผ่านหรือไม่ ถ้าต้องกดก็จะให้ทำการป้อนรหัสผ่านลงไป จากนั้นก็จะทำการ set buffer ส่งข้อมูล เพื่อส่งข้อมูลบัตรนี้ ไปให้ Center โดยผ่านทางคำสั่ง D3H และให้ Center ทำการตรวจสอบและส่งผลที่ตรวจสอบได้กลับมา โดยผ่านทางคำสั่ง D8H

คำสั่งที่ Center ส่งมาให้ Terminal นั้น มีคำสั่ง D0, D2, D4, D6, D8 ซึ่งได้กล่าวไว้แล้วในบทที่ 3 ซึ่งในส่วนนี้จะแสดงการทำงานของ Terminal เมื่อได้รับคำสั่งเหล่านี้ คือ

1. คำสั่ง D0 แสดงได้ดัง Flow Chart ในรูปที่ 4.7

จากรูปที่ 4.7 เมื่อ Terminal ได้รับคำสั่ง D0 จาก Center ซึ่งเป็นคำสั่งที่ส่งข้อมูลของเวลามาให้ Terminal จะทำการตรวจสอบเวลาที่รับได้นี้ว่าตรงกับ buffer ที่เก็บค่าเวลาของตัวเองหรือไม่ ถ้าไม่ตรงกันแสดงว่าเกิดการเปลี่ยนแปลงเวลา ถ้า Terminal ไม่มีการรูดบัตรในขณะนั้นก็จะทำการแสดงค่าเวลาใหม่ออกทาง LCD จากนั้นจะทำการตรวจสอบการเปิดประตู ถ้ามีการเปิดประตูอยู่ก็จะทำการลดค่าเวลาที่เปิดประตู และตรวจสอบว่าหมดเวลาของการเปิดประตูหรือยัง ถ้าหมดเวลาแล้วก็จะทำการปิดประตูและปิดเสียงเพลง และทำการเซ็ทบิทในไบต์สถานะที่กำหนดการกดรหัสผ่านในกรณีของ Mode 3 ให้ทำการกดรหัสผ่าน จากนั้นจะทำการลดค่าเวลาของการเปิดเครื่อง (เมื่อไม่มีการรูดบัตรเพื่อผ่านเข้าออก) ถ้าหมดเวลาของการเปิดเครื่องแล้ว ก็จะทำการส่งเอาท์พุทไปทำการปิดเครื่อง (อยู่ในสภาวะ Standby)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. คำสั่ง D2 เมื่อได้รับคำสั่งนี้ Terminal จะทำการส่งข้อมูลที่อยู่ใน buffer ส่งข้อมูลในขณะนั้นไปให้ Center ซึ่งก็คือเป็นการตอบกรับการ Poll ข้อมูลจาก Center

3. คำสั่ง D4 เมื่อได้รับคำสั่งนี้ Terminal จะทำการส่งข้อมูลที่อยู่ใน buffer ส่งข้อมูลในขณะนั้นไปให้ Center และจะทำการเซ็ทบิทในไบต์สถานะเพื่อแสดงการรอข้อมูลจากCenter ซึ่งก็คือเป็นคำสั่งที่จะให้ Terminal รอรับข้อมูลจาก Center

4. คำสั่ง D6 เมื่อได้รับคำสั่งนี้ Terminal จะทำการเซ็ท Mode การใช้งานของตัวเอง ตามค่าที่ส่งมาให้ ซึ่งใน Mode การใช้งานนี้แสดงอยู่ในหัวข้อของ การทำงานของระบบในตอนต้นของบทนี้ โดยจะต้องทำการเซ็ทบิทในไบต์สถานะดังนี้

- Mode 1 บิทแสดง Mode มีค่าเป็น "0" และ บิทแสดงการกดรหัสผ่านมีค่าเป็น "0"
- Mode 2 บิทแสดง Mode มีค่าเป็น "0" และ บิทแสดงการกดรหัสผ่านมีค่าเป็น "1"
- Mode 3 บิทแสดง Mode มีค่าเป็น "1" และ บิทแสดงการกดรหัสผ่านมีค่าเป็น "1"

5. คำสั่ง D8 เป็นการตอบรับข้อมูลของบัตรที่ Terminal ส่งไปให้ Center ทำการตรวจสอบและ Center ส่งผลของการตรวจสอบกลับมาผ่านทางคำสั่งนี้ เมื่อได้รับคำสั่งนี้ Terminal จะทำตามค่าที่ Center ส่งกลับมา ซึ่งประกอบด้วย

- สามารถผ่านเข้าไปได้ Terminal จะทำการส่งเอาท์พุทไปเปิดประตู และสร้างเสียงเพลงที่แสดงการเปิดประตู และแสดงผลว่าสามารถผ่านได้ (Pass OK) และแสดงผลว่าบันทึกเวลาเข้าทำงาน (Work in OK) เมื่อ Center บันทึกเวลาเข้าทำงานให้ และจะทำการรีเซ็ท Buffer ส่งข้อมูลให้อยู่ในสภาวะที่ไม่มีข้อมูลส่ง (คำสั่ง D5H)
- ไม่สามารถผ่านเข้าไปได้ จะทำการแสดงผลว่าไม่สามารถผ่านเข้าไปได้ (No Pass) และรีเซ็ท Buffer ส่งข้อมูลให้อยู่ในสภาวะที่ไม่มีข้อมูลส่ง
- รหัสผ่านไม่ตรง จะทำการลดจำนวนครั้งของการกดรหัสผ่านผิด ถ้ากดรหัสผ่านผิดครบ 3 ครั้งแล้ว ก็จะแสดงข้อความว่า ไม่สามารถผ่านได้ (No Pass) แต่ถ้ายังกดรหัสผิดไม่ครบ 3 ครั้ง จะให้มีการกดรหัสผ่านใหม่ และเซ็ท Buffer ส่งข้อมูลให้ส่งข้อมูลของบัตรใบนั้นและรหัสผ่านที่กดใหม่ไปให้ Center ทำการตรวจสอบใหม่อีก (ผ่านทางคำสั่ง D3)
- บันทึกเวลาออกจากงาน ถูกต้องหรือไม่ จะแสดงผล Work out ok เมื่อมีการบันทึกเวลาออกจากงานให้ และจะแสดงผล Work out missing เมื่อไม่มีการบันทึกเวลาให้ และจะทำการรีเซ็ท Buffer ส่งข้อมูลให้อยู่ในสภาวะที่ไม่มีข้อมูลส่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

โปรแกรมจัดการฐานข้อมูล
และโปรแกรมคิดเงินเดือนอัตโนมัติ

โปรแกรมที่เราใช้ในการสร้างฐานข้อมูลนี้เราจะใช้ภาษา C เป็นเทอร์โบซีของบอร์แลนด์ เวอร์ชั่น 2 ซึ่งเป็นภาษาที่เหมาะสำหรับการสร้างฐานข้อมูลแบบพื้นฐาน โดยมีคำสั่งที่ใช้กับฐานข้อมูล และการจัดการในเรื่องของตัวแปรในฐานข้อมูลอย่างง่าย ซึ่งเราจะแบ่งอธิบายโปรแกรมออกเป็นสองส่วนด้วยกันคือ ส่วนของโปรแกรมจัดการฐานข้อมูล และโปรแกรมคิดเงินเดือนอัตโนมัติ

โปรแกรมจัดการฐานข้อมูล

ฐานข้อมูลที่เราจะใช้ในการเก็บข้อมูล เราจะกำหนดให้มีขนาดที่ใช้กับพนักงาน 100 คน โดยใช้เก็บ ชื่อ , นามสกุล , เพศ , อายุ , เงินเดือน , ที่อยู่ , ตำแหน่ง , วันเดือนปีเกิด , เลขบัญชีธนาคาร เลขประจำตัวประชาชน และ เลขที่บัตรประกันสังคม ซึ่งเป็นสิ่งสำคัญที่จะใช้ในการเก็บประวัติของพนักงานในบริษัท

โดยภาษาซีมีตัวแปรที่ชื่อว่า STRUCTURE ไว้สำหรับเก็บข้อมูลประเภทนี้โดยตรงโดยจะกำหนดเนื้อที่สำหรับเก็บข้อมูลแต่ละคนออกเป็น RECORD โดยแต่ละ RECORD จะแบ่งออกเป็น ส่วน FIELD ย่อยเอาไว้สำหรับเก็บข้อมูลพนักงานแต่ละคนเช่น ชื่อ , ที่อยู่ ฯลฯ ซึ่งเราจะกำหนดให้

1. ชื่อ และ นามสกุล มีขนาด 35 ไบต์
2. เพศ มีขนาด 6 ไบต์ (ชาย , หญิง)
3. อายุ มีขนาด 2 ไบต์
4. เงินเดือน มีขนาด 4 ไบต์
5. ที่อยู่ มีขนาด 70 ไบต์
6. ตำแหน่ง มีขนาด 20 ไบต์
7. วันเดือนปีเกิด มีขนาด 20 ไบต์
8. เลขที่บัญชีธนาคาร มีขนาด 20 ไบต์
9. เลขบัตรประจำตัวประชาชน มีขนาด 20 ไบต์
10. เลขบัตรประกันสังคม มีขนาด 20 ไบต์
11. กรุ๊ปเลือด มีขนาด 3 ไบต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยจะรวมกันแต่ละ RECORD จะมีขนาด 220 ไบต์ต่อพนักงาน 1 คน ซึ่งถ้าเรากำหนดให้มีจำนวนพนักงาน 100 คน ก็จะมีขนาดไฟล์เท่ากับ 22000 ไบต์ โดยใน PROJECT นี้เราได้เก็บฐานข้อมูลของพนักงานแต่ละคน เอาไว้ในไฟล์ที่ชื่อว่า " PERSON.DAT " โดยจะมีโปรแกรมที่ใช้ในการจัดการฐานข้อมูลชื่อว่า " DBMAIN.EXE " โดยจะเรียกโปรแกรมฐานข้อมูล PERSON.DAT มาเก็บไว้ใน MEMORY ทั้งหมด ซึ่งจะทำให้การแก้ไขหรือ เรียกใช้เป็นไปได้อย่างรวดเร็วโดยโปรแกรมที่ใช้ในจัดการฐานข้อมูลที่มีอยู่ในโปรแกรม DBMAIN.EXE จะมีอยู่ด้วยกัน 3 ส่วนคือ ส่วนของการแสดงผล ส่วนของการแก้ไขข้อมูลพนักงาน และ ส่วนของการ_ลบข้อมูลออก

โปรแกรมที่ใช้นี้เราจะใช้งานร่วมกับโปรแกรมภาษาไทย เพื่อที่จะสามารถแสดงผลและเก็บข้อมูลเป็นภาษาไทยได้ โดยจะใช้โปรแกรมภาษาไทยใน DOS เวอร์ชั่น 6.0 THAI EDITION ซึ่งมีโปรแกรมสำหรับการใช้ภาษาไทยอยู่ ซึ่งมีชื่อว่า THAILS.EXE ซึ่งก่อนที่เราจะทำการ RUN โปรแกรม DBMAIN.EXE เราจะต้องทำการโหลดโปรแกรมภาษาไทยเสียก่อนแล้วจึงจะใช้งานได้ โดยการเซ็ทค่า CONFIGURE ของภาษาไทย สามารถทำได้โดย RUN โปรแกรม THAIMAN.EXE โดยเขียนลงใน DOS PROMPT ว่า THAIMAN/S

ฐานข้อมูลที่เราสร้างขึ้นนี้จะใช้ POINTER เป็นตัวกำหนดตำแหน่งข้อมูลของพนักงาน โดยเรียงลำดับกันไปตั้งแต่ 1 ถึง 100 ซึ่งจะเป็นรหัสพนักงานเดียวกันกับรหัสพนักงานที่อยู่ในเครื่อง CENTER ซึ่งมีหน้าที่เก็บข้อมูลของแต่ละบัตรเอาไว้ และเก็บค่าเวลาการทำงานเพื่อส่งให้คอมพิวเตอร์ในแต่ละวัน โดยที่คอมพิวเตอร์จะไม่รู้รหัสภายในบัตร และรหัสผ่านซึ่งมีใน CENTER เลยเพื่อป้องกันการลักลอบดูข้อมูลหรือรหัสผ่านของแต่ละคนได้

โปรแกรมคิดเงินเดือนอัตโนมัติ

โปรแกรมคิดเงินเดือนอัตโนมัติจะนำข้อมูลที่ได้จาก CENTER มาคำนวณหาค่าเวลาที่มาทำงานในแต่ละวัน โดยข้อมูลจะถูกส่งผ่านทางพอร์ตอนุกรม RS - 232 เข้ามายังคอมพิวเตอร์โดยจะถูกเก็บไว้เป็นไฟล์เพื่อนำไปใช้คำนวณต่อๆไปโดยเราจะแยกการทำงานออกเป็นสองส่วนดังนี้

การติดต่อข้อมูลกับ Center

การติดต่อกับ Center คือ Center จะทำการส่งข้อมูลเวลาเข้าออก ของพนักงานมาให้เครื่องคอมพิวเตอร์ เพื่อให้คอมพิวเตอร์ทำการเก็บข้อมูลไว้ และคำนวณเป็นเงินเดือนออกมา ซึ่งในการติดต่อข้อมูลในส่วนนี้ จะติดต่อผ่านทางมาตรฐาน RS-232 โดย Center จะทำการเปิดเครื่องคอมพิวเตอร์โดยอัตโนมัติ เมื่อต้องการส่งข้อมูลให้คอมพิวเตอร์ โดยโปรแกรมในส่วนของเครื่องคอมพิวเตอร์ที่จะทำการรับข้อมูล จะอยู่ในส่วนของ Autoexec.bat ซึ่งเป็นแบทช์ไฟล์ที่จะทำงานเมื่อมีการเปิดเครื่องคอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ ทำซ้ำ ดัดแปลง เผยแพร่ หรือกระทำการใดๆ ในลักษณะที่เป็นการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หรือเมื่อมีการรีเซ็ท ซึ่งโปรแกรมที่จะใช้ในส่วนนี้จะประกอบไปด้วย

1. ส่วนของการรับข้อมูลจาก Center และเก็บไว้ในไฟล์ Buffer.asc

2. ส่วนของการนำข้อมูลใน Buffer.asc มาต่อท้ายไฟล์ในเดือนปัจจุบัน

3. ส่วนของการตรวจสอบวันที่ 1 ของทุกๆเดือน เพื่อทำการแปลงไฟล์ข้อมูลในเดือนที่ผ่านมาเป็น ไฟล์ของตัวเลขที่สามารถนำไปคำนวณค่าเวลาการทำงานได้

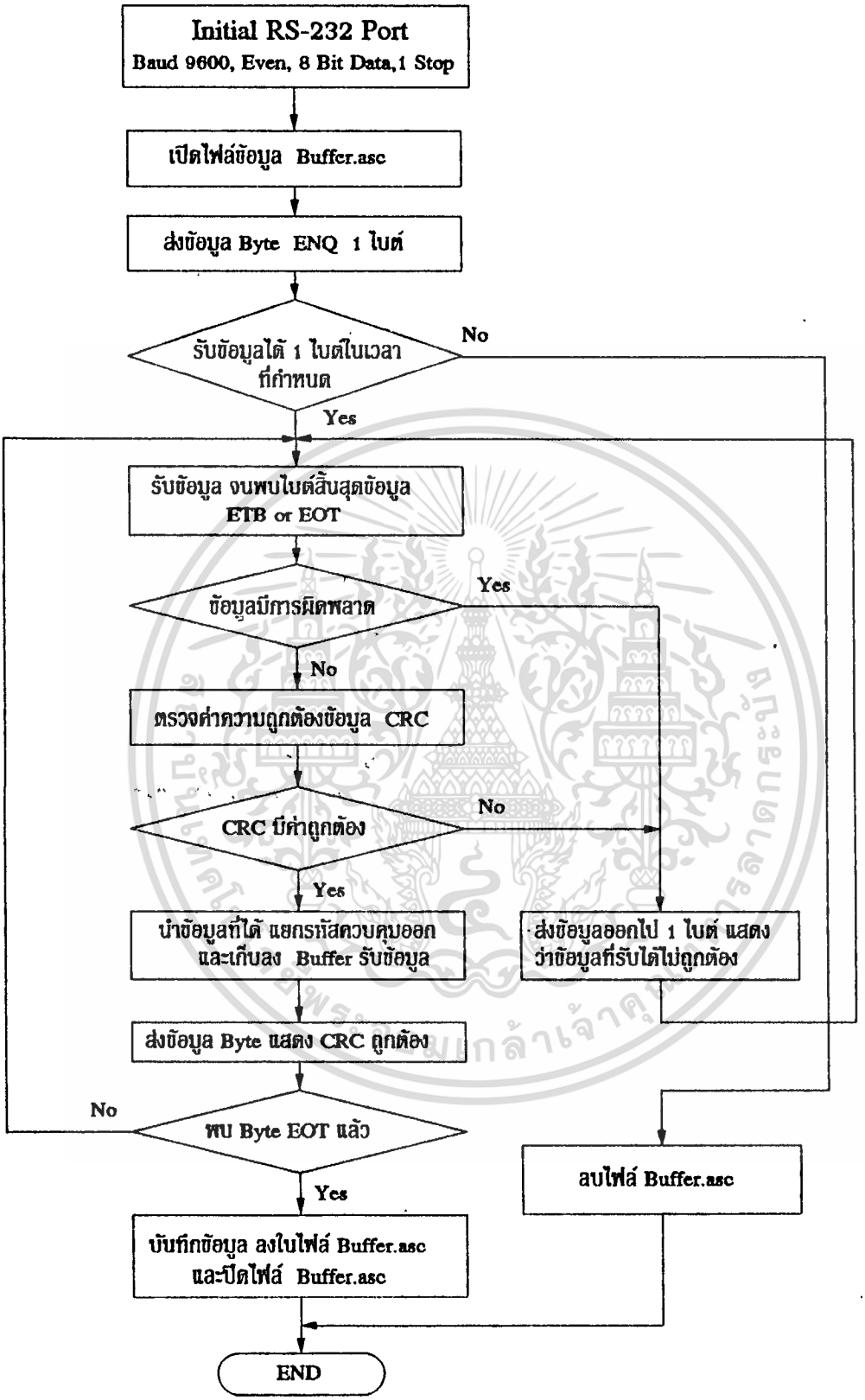
1. ส่วนของการรับข้อมูลจาก Center

ในการส่งข้อมูลจาก Center มาให้คอมพิวเตอร์นั้น จะมีรูปแบบในการส่งข้อมูลดังที่ได้แสดงไว้แล้วในบทที่ 4 ในส่วนของการทำงานของ Center ในหัวข้อการติดต่อกับคอมพิวเตอร์ ซึ่งจะเห็นว่าในการรับและส่งข้อมูล จะมีการส่งข้อมูลออกเป็นบล็อก โดยจะมีการตรวจสอบความถูกต้องของข้อมูลในแต่ละบล็อก ซึ่งโปรแกรมส่วนที่จะรับข้อมูลจาก Center นี้ จะแสดงได้ดัง Flow Chart ในรูปที่ 5.1

จากรูปที่ 5.1 การรับข้อมูลจาก Center จะเริ่มจาก ทำการเซ็ทค่าพอร์ตที่จะทำการติดต่อกับ Center ก่อน (ใช้พอร์ต COM2) จากนั้นจะทำการเปิดไฟล์ที่จะใช้ในการบันทึกข้อมูล ต่อไปจะทำการส่งไบต์ ENQ (05H) ไปให้ Center เพื่อบอกให้ Center รู้ว่าขณะนี้คอมพิวเตอร์พร้อมที่จะรับข้อมูลแล้ว และให้ส่งข้อมูลมาให้ได้ เมื่อคอมพิวเตอร์รับข้อมูลจนพบไบต์ ETB หรือ EOT ซึ่งเป็นไบต์แสดงจุดสิ้นสุดของบล็อก หรือจุดสิ้นสุดของข้อมูล และจะทำการตรวจสอบข้อมูลที่รับมาได้ในบล็อกนั้นว่าถูกต้องหรือไม่ ถ้าถูกต้องก็จะส่งไบต์แสดงข้อมูลถูกต้อง กลับไปให้ Center และเตรียมรับข้อมูลบล็อกต่อไป ถ้ายังไม่พบ EOT แต่ถ้าข้อมูลบล็อกนั้นไม่ถูกต้อง ก็จะทำการส่งไบต์แสดงข้อมูลผิดกลับไปให้ Center เพื่อจะทำการรับข้อมูลบล็อกที่ผิดนั้นใหม่ เมื่อการรับข้อมูลเป็นไปอย่างถูกต้องจนครบทั้งหมด ก็จะทำการนำข้อมูลที่รับได้ทั้งหมดบันทึกลงไฟล์ที่เปิดขึ้น เพื่อนำข้อมูลในไฟล์นี้ไปใช้งานต่อไป แต่ถ้าในการรับข้อมูลไม่สมบูรณ์ หรือไม่มีข้อมูลจาก Center ส่งมาให้ (กรณีที่เป็นการเปิดเครื่องคอมพิวเตอร์ตามปกติ) ก็จะทำการลบไฟล์ที่เปิดไว้ในครั้งแรกออก เพื่อให้รู้ว่าไม่มีข้อมูลส่งมาจาก Center

ค่าที่ได้จากการส่งมาทางพอร์ต RS-232 นั้น จะเป็นรหัส ASCII ซึ่งเราจะเก็บค่ารหัส ASCII นั้นเอาไว้ในไฟล์ Buffer.asc ก่อนซึ่งค่าดังกล่าวเราจะไม่สามารถนำไปคำนวณได้ เราจะต้องทำการแปลงค่าดังกล่าวไปเป็นค่า Numeric หรือ ค่าตัวเลขเสียก่อนจึงจะนำไปใช้งานจริงได้ซึ่งจะกล่าวในหัวข้อต่อๆไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.1 Flow Chart แสดงโปรแกรมรับข้อมูลจาก Center

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. ส่วนของการนำข้อมูลที่ได้ไปต่อท้ายในไฟล์เดือนปัจจุบัน

ข้อมูลที่เก็บไว้ในไฟล์ Buffer.asc จะเป็นข้อมูลในแต่ละวันซึ่งไฟล์ที่ทำ การรวบรวมข้อมูลของทุก ๆ วัน จะมีชื่อไฟล์เป็นชื่อย่อของเดือนภาษาอังกฤษ 3 ตัวและ ตัวเลขท้ายปีพุทธศักราช 2 ตัวและมีนามสกุลเป็น " .asc " เช่นในเดือน มกราคม 2537 ก็จะมีชื่อไฟล์ในเดือนนี้ว่า " jan37.asc " ในเดือนกุมภาพันธ์ 2539 ก็จะมีชื่อไฟล์ในเดือนนี้ว่า " feb39.asc " เป็นต้น โดยข้อมูลในไฟล์แต่ละวันจะถูกนำมาบันทึกต่อท้ายในไฟล์แต่ละเดือน โดยเรียงข้อมูลต่อท้ายกันไปเรื่อยๆ

การทำงานของโปรแกรมนี้ก็จะทำการเปิดไฟล์ในชื่อไฟล์เดือนนั้น ๆ โดยหากว่ายังไม่มีชื่อไฟล์นั้น (ในวันที่ 1 ของทุกๆเดือน) ก็จะทำการสร้างไฟล์ขึ้นมาใหม่แล้วทำการนำเอาข้อมูลในแต่ละวันมาเก็บไว้ในไฟล์เดือนปัจจุบัน เมื่อทำการเก็บข้อมูลเสร็จแล้วเราจะทำการลบไฟล์ Buffer.asc นั้นออกไปเพื่อที่จะทำการบันทึกค่าเวลาในวันต่อๆไป

3. ส่วนของการตรวจสอบวันที่ 1 ของทุกๆเดือน

ข้อมูลที่ได้จากการนำเข้ามาทางพอร์ตอนุกรม RS-232 ซึ่งได้ถูกเก็บไว้ในไฟล์ของในเดือนต่าง ๆ แล้วจะเป็นรหัส ASCII ซึ่งจะไม่สามารถนำไปคำนวณเวลาการทำงานได้ เราจะต้องทำการแปลงให้เป็นค่าตัวเลขก่อนโดยจะทำการแปลงค่า ASCII ไปเป็นค่าตัวเลข (Numeric) เพื่อที่จะได้นำไปคำนวณค่าเวลาการทำงานของพนักงาน

โปรแกรมที่เราสร้างขึ้นจะทำการตรวจสอบว่าเวลาในขณะที่โปรแกรมทำงานนั้นเป็นวันที่เท่าใด ซึ่งถ้าไม่ใช่วันที่ 1 แล้วโปรแกรมก็จะออกจากการทำงานไป ที่เราใช้วันที่ 1 นี้เนื่องจากเป็นวันที่เราจะต้องจ่ายเงินเดือนให้กับพนักงาน และเป็นวันที่เราสามารถตรวจสอบได้ง่ายกว่าวันสิ้นเดือน เนื่องจากมีค่าวันเวลาที่ไม่แน่นอน เช่น เดือนที่ลงท้ายด้วย"คม" ก็จะเป็นวันที่ 31 แต่หากลงท้ายด้วย"ยน" ก็จะเป็นวันที่ 30 ดังนั้นเราจึงเลือกเอาวันที่ 1 ซึ่งง่ายต่อการคำนวณมาก แต่ก็มีข้อเสียบ้างในการจ่ายเงินเดือนซึ่งจะต้องจ่ายเงินเดือนหลังเวลาที่เรากำหนดไว้ของวันที่ 1 ซึ่งมักจะเป็นเวลาที่พนักงานเลิกงานแล้ว จึงจำต้องจ่ายเงินเดือนในวันที่ 2 เป็นต้นไป

การทำงานของโปรแกรมจะเริ่มเมื่อวันที่โปรแกรมทำงานในขณะนั้นเป็นวันที่ 1 ของทุก ๆ เดือน โปรแกรมจะนำข้อมูลรหัส ASCII ของในเดือนที่ผ่านมา มาทำการแปลงเป็นค่าตัวเลขซึ่งสามารถเขียนได้เป็นสมการดังนี้

NUM = [(SNUM [0] - '0') * 10] + (SNUM [1] - '0')

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

NUM เป็นค่าตัวเลขที่ได้จากการแปลงรหัส ASCII มีขนาด 1 ไบต์ (มีค่าได้ไม่เกิน 128)

SNUM เป็นรหัส ASCII โดย [0] เป็นไบต์สูง (หลักสิบ) และ [1] เป็นไบต์ต่ำ (หลักหน่วย) มีขนาด 2 ไบต์

สมมติว่าค่าใน SNUM[0] = 31 และ SNUM[1] = 37 สมการของการคำนวณก็จะได้ค่าของการรวมกันระหว่าง 10+7 เท่ากับ 17 นั่นเอง ค่าที่ได้จากการคำนวณจะนำไปเก็บไว้ในไฟล์ที่เป็นชื่อย่อของเดือนภาษาอังกฤษ 3 ตัว และตัวเลขท้ายปีพุทธศักราช 2 ตัวและนามสกุล " .dat "เช่นในเดือนมกราคม 2537 ก็จะมีชื่อไฟล์ในเดือนนี้ว่า " jan37.dat "โดยค่าที่ได้จากการคำนวณจะถูกเก็บไว้ในไฟล์ดังกล่าว แล้วทำการคำนวณค่ารหัส ASCII ตัวต่อไปจนหมดไฟล์

เมื่อทำการแปลงค่าเสร็จเรียบร้อยแล้ว เราจะทำการเก็บชื่อไฟล์เดือนที่ทำการแปลงไว้ในไฟล์ที่ชื่อว่า " Month.lib " ซึ่งเป็นไฟล์ที่รวบรวมชื่อของไฟล์เดือนที่ทำการแปลงข้อมูลเรียบร้อยแล้วเพื่อความสะดวกในการเรียกใช้ไฟล์ดังกล่าว

นอกจากนี้ตัวโปรแกรมยังมีการป้องกันการผิดพลาด เนื่องจากการที่ มีการเปลี่ยนค่าพุทธศักราชไปเป็น พ.ศ.ใหม่ โดยการตรวจสอบวันที่ 1 ของเดือนมกราคม ของทุก ๆ ปีให้มีการทำการแปลงไฟล์ให้เป็นเดือนธันวาคมของ พ.ศ. ที่ผ่านมาได้ ซึ่ง พ.ศ.ที่สามารถใช้งานโปรแกรมนี้ สามารถใช้ได้ถึง พ.ศ. 2560 ซึ่งเพียงพอในการใช้งานมากอยู่แล้ว

<u>ส่วนของการคำนวณเงินเดือนอัตโนมัติ</u>

ก่อนที่เราจะไปทำความเข้าใจในวิธีการคิดเงินเดือนเรามาดูหลักเกณฑ์ของการกำหนดเวลาและการคิดเงินเดือนโดยกำหนดไว้เป็นข้อๆดังนี้

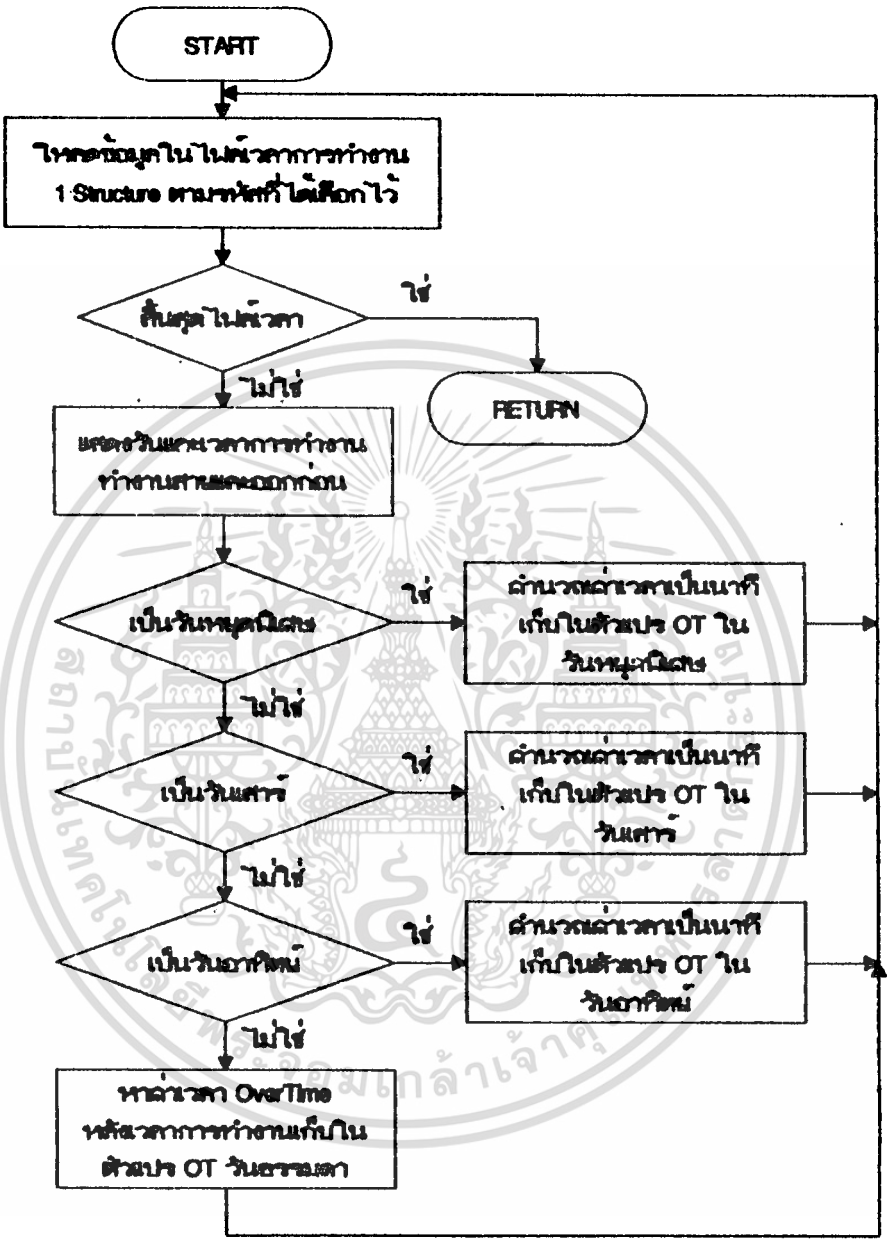
<u>หลักเกณฑ์การคิดเงินเดือน</u>

1. การตรวจสอบการมาสายจะกำหนดเวลาการมาสายได้ไม่เกินกำหนดเป็นนาทีถ้ามาสายเกินกำหนด จะนำค่าเวลาที่มาสายทั้งเดือนนั้นรวมกัน แล้วหักออกด้วยกำหนดเวลาการมาสายแล้วนำมาคูณกับเปอร์เซนต์ที่กำหนดไว้ในการหักเงินเดือน ถ้ามาสายเกินกำหนดเวลา ไปลบออกจากเงินเดือนเช่น กำหนดห้ามมาสายเกิน 100 นาที มิฉะนั้นจะหัก 0.001% สมมติ นาย ก.เงินเดือน 10000 บาท เกิดมาสายรวมกันทั้งเดือน 110 นาที ก็จะได้เท่ากับ 10000 - (10000 * 0.001 / 100 * (110 - 100)) = 9999 หมายความว่า นาย ก. จะถูกหักเงินเดือนไป 1 บาท

2. การตรวจสอบการออกก่อนเวลาจะไม่มีผลในการคำนวณเงินเดือนแต่จะแสดงไว้ในการดู เวลาการทำงานเพื่อพิจารณาการเลื่อนขั้นหรือขึ้นเงินเดือน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมเดือนเดือน



รูปที่ 5.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. การทำ Over Time ในวันหยุดพิเศษ , วันเสาร์ และ วันอาทิตย์ จะคิดเป็นนาทีโดยจะมีการคำนวณเป็นค่าเวลาที่ทำ OT ในแต่ละประเภท มาคูณกับจำนวนเท่าที่กำหนดไว้แล้วนำไปคูณกับค่าเงินเดือนต่อนาที (คิดจากเงินเดือนหารด้วย 30 วัน หารด้วย 8 ช.ม. และหารด้วย 60 นาที)

4. การมาก่อนเวลาจะคิดที่เวลาที่กำหนดเริ่มทำงานโดยที่หากมาสายจะถูกหักในข้อ 1 แล้วยังคิดเริ่มทำงานที่เวลามาสาย (เฉพาะในวันที่เป็น OverTime) โดยวันธรรมดาจะมีผลเพียงข้อ 1

5. การทำ OverTime ในวันธรรมดา จะเริ่มคิดเวลาที่กำหนดเริ่มการทำ OverTime และถ้าหากออกก่อนเวลาออก OverTime จะคิดที่เวลาออก แต่ถ้าออกหลังเวลาออก OverTime จะคิดที่เวลาออก Over Time ที่กำหนด

6. วันที่พนักงานขาดงานหรือไม่มาทำงานในวันธรรมดาหรือวันที่กำหนดให้มาทำงาน จะไม่นำมาคิดเงินเดือน โดยพนักงานจะต้องส่งใบลากิจ , ใบลาป่วย , ใบลาพักร้อนอยู่แล้ว ซึ่งจะไม่มีผลต่อการหักเงินเดือนของพนักงาน

<u>การคำนวณเงินเดือนพนักงาน</u>

การคำนวณเงินเดือน จะมีหลักเกณฑ์โดยเอาเงินเดือนของพนักงานหักออกด้วยเกณฑ์การมาสายแล้วบวกด้วยเงินที่ได้จากการทำ OverTime ในวันหยุดและในเวลาพิเศษของวันธรรมดา โดยการคำนวณเวลาการทำ OverTime แสดงในรูปของโฟลว์ชาร์ทการคิดเงินเดือนซึ่งจะแยกตัวแปรของการทำ Over Time ออกเป็น 4 ตัว คือ OT ในวันหยุดพิเศษ , OT ในวันเสาร์ , OT ในวันอาทิตย์ , OT ในวันธรรมดา ซึ่งค่าเวลาที่ได้จะนำมาคำนวณร่วมกับค่าที่กำหนดไว้ในไฟล์ " DBMAIN.CFG " ซึ่งรวบรวมกำหนดเวลาการทำงาน , การทำ OverTime, จำนวนเท่าที่จ่ายในการทำ OverTime ต่างๆ ซึ่งเราสามารถกำหนดได้โดยเรียกโปรแกรม DBMAIN แล้วเลือกหัวข้อกำหนดค่า CONFIGURE ซึ่งจะสามารถทำการกำหนดค่าเวลาการทำงานต่าง ๆ เองได้ อีกทั้งยังกำหนดวันหยุดในแต่ละปีได้อีกด้วย ซึ่งอยู่ในโปรแกรม DBMAIN เช่นเดียวกัน

ค่า CONFIGURE ของโปรแกรมที่ถูกเก็บไว้ใน ไฟล์ DBMAIN.CFG มีรายละเอียดซึ่งแบ่งออกได้เป็น 3 ส่วนดังนี้

1. ส่วนของการกำหนดเวลา

จะกำหนดเวลาการทำงาน เข้า - ออก ของแต่ละวัน และเวลาเข้าและออกของการทำ OverTime ซึ่งสามารถกำหนดให้มีเวลาการทำงานตามกำหนดซึ่งสามารถนำไปคำนวณเงินเดือนใหม่ได้ทันที เนื่องจากการคำนวณเงินเดือนจะทำพร้อมกับไฟล์ DBMAIN.CFG และ การแสดงผลออกทางหน้าจอ

2. ส่วนของการกำหนดอัตราการตอบแทนการทำงานในเวลา OverTime เป็นจำนวนเท่า

จะเซ็ทค่าจำนวนเท่าของการทำ OverTime ในแต่ละประเภท ซึ่งแยกออกจากกัน คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จำนวนเท่าของการทำ OverTime ในวันหยุดพิเศษ , OverTime ในวันเสาร์ , OverTime ในวันอาทิตย์ และ OverTime ในวันธรรมดา (ในเวลาหลังจากเวลาการทำงาน) ซึ่งจะกำหนดเป็นจำนวนเท่าที่เป็นเลขทศนิยมก็ได้

3. ส่วนของการกำหนดการมาสายและการหักเงินเดือน

จะกำหนดค่าเวลาการมาสายไม่เกินเป็นจำนวนนาที (สามารถกำหนดได้ไม่เกิน 128) ซึ่งถ้าหากว่ามาสายเกินเวลาจะมีการหักเงินเดือนเป็นจำนวนเปอร์เซนต์ของเงินเดือนต่อนาทีซึ่งได้กรอกไว้แล้วในหลักเกณฑ์การคิดเงินเดือน ในข้อที่ 1

ปัญหาที่เกิดขึ้นในการเขียนโปรแกรมภาษาซี

ปัญหาที่เกิดขึ้นในโปรแกรมมักจะเกิดจากการเขียนโปรแกรมแล้วไม่สามารถนำไปใช้งานได้ซึ่งมักจะพบมากในโปรแกรมภาษาซี ซึ่งเราจะไม่กล่าวถึง โดยการบอกกล่าวถึงปัญหานี้จะช่วยในการพัฒนาโปรแกรมให้มีประสิทธิภาพยิ่งขึ้น ซึ่งปัญหาต่างๆสามารถสรุปเป็นข้อๆได้ดังนี้

1. ปัญหาการใช้ตัวแปรโครงสร้าง (STRUCTURE) ถ้าหากนำไปประยุกต์ใช้กับไฟล์แบบไบนารี่โดยใช้คำสั่ง fseek จะสามารถทำให้การค้นหามีประสิทธิภาพยิ่งขึ้น แต่การใช้คำสั่ง fseek นี้ในโปรแกรมภาษาซี มักจะมีปัญหาของข้อมูลหลัง POINTER ที่ fseek ไปชี้ตำแหน่งอยู่หายไปทำให้ไม่สามารถใช้การเก็บไฟล์แบบไบนารี่ได้ ซึ่งการเก็บไฟล์แบบไบนารี่จะมีข้อดีมากกว่า เช่น การเก็บข้อมูลพนักงานไม่จำเป็นต้องระบุจำนวนพนักงานสามารถเก็บข้อมูลได้จำนวนไม่จำกัดขึ้นอยู่กับขนาดของดิสก์ ซึ่งถ้าเป็นฮาร์ดดิสก์ก็จะสามารถเก็บเป็นจำนวนมากๆได้ และการค้นหาข้อมูลสามารถทำได้อย่างรวดเร็วและมีรูปแบบมากยิ่งขึ้น

2. ปัญหาการใช้โปรแกรมภาษาไทยใน DOS 6.0 THAI EDITION ซึ่งมีปัญหาการจัดรูปแบบตัวอักษร ทำให้การแสดงผลเป็นไปได้อย่างไม่ค่อยดีนัก เกิดช่องว่างของตัวอักษรเมื่อมีการใช้สระที่อยู่ตำแหน่งเดิม จึงอยากจะแนะนำให้หาโปรแกรมจัดการระบบภาษาไทยที่ดีกว่า

3. ปัญหาการกำหนดหลักเกณฑ์การคำนวณเงินเดือน เนื่องจากมีหลายบริษัทที่มีวิธีการใช้การคำนวณที่แปลกออกไป แตกต่างกันไป ซึ่งไม่สามารถนำมาประยุกต์รวมกันได้

4. ปัญหาความไม่แน่นอนของข้อมูล ซึ่งในระบบที่เราได้ออกแบบไว้นั้นถ้าหากเกิดมีการสูญหายของข้อมูลเพียงตัวใดตัวหนึ่ง จะทำให้เกิดการผิดพลาดของข้อมูลทั้งหมด เช่น หากเกิด BAD SECTOR ที่ตำแหน่งข้อมูลของเวลาการทำงานจะทำให้ข้อมูลเกิดการผิดพลาดตั้งแต่จุดที่เกิดการสูญเสียข้อมูลจนจบไฟล์นั้นๆ ซึ่งหากเป็นไปได้คงจะต้องมีโปรแกรมคอยตรวจสอบข้อมูลและสามารถแก้ไขได้ซึ่งจะต้องพัฒนากันต่อไปอีก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6

การใช้เครื่อง

1. การใช้เครื่อง Center

ในการใช้ส่วนของเครื่อง Center จะมีการใช้งานผ่านทาง Key SW ซึ่งมี Key ที่ใช้งาน ดังนี้

- Key Number 0-9 ไว้ใช้ในการป้อนค่าต่าง ๆ ที่เป็นตัวเลขให้เครื่อง
- Key Enter ใช้เริ่มการทำงาน หรือป้อนค่าให้ให้เครื่อง
- Key Function ใช้ในการเริ่มการทำงานของฟังก์ชั่นต่าง ๆ
- Key ESC ใช้ในการยกเลิกการทำงานในขณะนั้น (ยกเลิกทีละขั้นตอน)
- Key Del ใช้ในการลบค่าไป 1 ตัว เมื่อทำการใส่ค่าผิด
- Key Left,Right ใช้ในการเลือกฟังก์ชั่นต่าง ๆ
- Key Yes,No ใช้ในการตกลง หรือไม่ตกลง

การเลือกฟังก์ชั่นการใช้งาน

ฟังก์ชันที่ใช้งานนี้มี 4 ฟังก์ชั่น การเลือกฟังก์ชั่นต่าง ๆ ทำได้โดย

- กดคีย์ Function
- กดคีย์ number 1-4 เพื่อเลือกฟังก์ชั่น 1-4 และกดคีย์ Enter หรือ กดคีย์ Enter ก่อน และใช้คีย์ Left,Right เลือกฟังก์ชั่นที่ต้องการ และกดคีย์ Enter อีกครั้ง

โดยในการใช้ฟังก์ชั่นต่าง ๆ จะแสดงได้ดังนี้

1.1 การกำหนดค่าเวลาเปิด-ปิดเครื่อง และเวลาส่งข้อมูลให้คอมพิวเตอร์

- เลือก Function 1
- จากนั้น LCD จะแสดงข้อความให้ใส่ค่าเวลาเปิดเครื่อง (TIME:Open Center) และจะแสดงค่าเดิมให้เห็นด้วย ถ้าต้องการเปลี่ยนค่าใหม่ให้ใส่ค่าเวลาที่ต้องการลงไป โดยใส่ค่าเวลาในรูปแบบ 24 ชั่วโมง ติดกันไปทั้ง 4 ตัว เช่น 0530 หมายถึงเวลา 5:30 น. แต่ถ้าไม่ต้องการเปลี่ยนแปลงค่า (ใช้ค่าเดิมที่แสดง) ก็ให้กดคีย์ Enter
- จากนั้น LCD จะแสดงข้อความให้ใส่ค่าเวลาปิดเครื่อง (TIME:OFF Center) ก็ให้ใส่ค่าเหมือนเวลาเปิดเครื่อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- จากนั้น LCD จะแสดงข้อความให้ใส่ค่าเวลาส่งข้อมูลให้คอมพิวเตอร์ (TIME: Send Data) ก็ให้ใส่ค่าเหมือนเวลาเปิดเครื่อง

1.2 การเพิ่มบัตรใหม่ หรือการแก้ไขบัตรเก่า

- เลือก Function 2

- LCD จะแสดงข้อความให้ใส่บัตรใหม่เข้าไป ให้ใส่บัตรเข้าไปในเครื่องอ่าน ถ้าบัตรนั้นใช้ไม่ได้ ก็จะแสดงข้อความว่า "Card Error" แต่ถ้าบัตรนั้นใช้ได้ ก็จะแสดงข้อความให้เลือกชนิดของบัตร ให้ใช้คีย์ Left, Right ในการเลือกชนิดของบัตร และกดคีย์ Enter

- ถ้าเลือกบัตรแบบตรวจสอบเวลาเข้าออก (Check Time:Type) เครื่องจะให้เลือกตัว Terminal ที่จะเพิ่มบัตรนี้เข้าไป หรือเป็น Terminal ตัวที่มีข้อมูลของบัตรใบนี้อยู่ ในกรณีที่จะแก้ไข โดยใส่เป็นตัวเลขประจำ Terminal (1-16) และกด Enter จากนั้นเครื่องจะตรวจสอบว่าบัตรใบนั้นเป็นบัตรเก่าหรือบัตรใหม่และจะแสดงผลให้ทราบด้วย จากนั้นให้ทำการใส่รหัสผ่านลงไปจำนวน 4 ตัว (Pass Code) ต่อด้วยใส่รหัสประจำตัว 4 ตัว (Person Code) ถ้าในการใส่รหัส หรือการใส่ค่าใด ๆ ผิด เครื่องจะให้ทำการใส่ค่านั้นใหม่ เมื่อเรียบร้อยแล้วเครื่องจะทำการเลื่อนบัตรออกมา และแสดงข้อความให้ใส่บัตรใหม่เข้าไป แต่ถ้าไม่ต้องการเพิ่มบัตรแล้ว ให้กดคีย์ ESC ออกจากฟังก์ชั่น

- ถ้าเลือกบัตรแบบไม่ตรวจสอบเวลาเข้าออก (Extra Type) เครื่องจะทำการตรวจสอบว่าบัตรใบนี้เป็นบัตรเก่า หรือบัตรใหม่ และแสดงผลให้ทราบด้วย จากนั้นให้ทำการใส่รหัส ทั้ง 2 แบบลงไป (เหมือนแบบ ตรวจสอบเวลาเข้าออก) จากนั้นเครื่องจะให้ทำการกำหนด Terminal ที่บัตรใบนี้จะสามารถผ่านเข้าไปได้ โดยจะถามทีละ Terminal ให้ทำการตอบ Yes เมื่อสามารถผ่านได้ และตอบ No เมื่อไม่สามารถผ่านได้ ให้ใส่ไปจนครบทั้ง 16 Terminal จากนั้น เครื่องจะเลื่อนบัตรออกมา และให้ทำการใส่บัตรใหม่เข้าไป ถ้าไม่ต้องการเพิ่มบัตรแล้ว ให้กดคีย์ ESC ออกจากฟังก์ชั่น

1.3 การลบบัตรที่ใช้ในการเข้าออก

- เลือก Function 3

- เครื่องจะให้ใส่ Person Code ของบัตรที่ต้องการจะลบ ให้ทำการใส่ Person Code ของบัตรใบที่ต้องการลบ ลงไป และกด Enter

- ถ้ารหัสนั้นตรงกับ รหัสประจำบัตรที่อยู่ในส่วนของ Extra Card ก็จะแสดงข้อความถามว่าต้องการลบบัตรใบนี้ที่อยู่ใน Extra Card หรือไม่ ให้ทำการตอบ Yes หรือ No

- ถ้ารหัสนั้นตรงกับ รหัสประจำบัตรที่อยู่ในส่วนของ บัตรแบบบันทึกเวลาเข้าออกของ Terminal ตัวใด ๆ ก็จะแสดงข้อความถามว่าต้องการลบบัตรใบที่อยู่ใน Terminal นั้น ๆ หรือไม่ ให้ทำการตอบ Yes หรือ No

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- เมื่อสิ้นสุดการลบข้อมูลของบัตรแต่ละใบ เครื่องจะให้ใส่ Person Code ที่ต้องการลบอีก เมื่อไม่ต้องการลบแล้วให้กดคีย์ ESC เพื่อออกจากฟังก์ชั่น

1.4 การเซ็ทค่าเวลาปัจจุบันให้เครื่อง

- เลือก Function 4

- เครื่องจะให้ใส่ค่าเวลาปัจจุบัน ซึ่งประกอบด้วย

- Day of week ให้ใส่ค่าวันในรอบสัปดาห์โดยเริ่มที่วันอาทิตย์ซึ่งมีค่า 0 และวันจันทร์จะมีค่า 1 และต่อไปเรื่อย ๆ จนถึงวันเสาร์ ซึ่งมีค่า 6

- Year ให้ใส่ค่าปี โดยใส่ค่าปีเป็น ค.ศ. โดยค่าสองตัวหลังของปี

- Month ให้ใส่ค่าเดือน

- Day ให้ใส่วันที่

- Hour ให้ใส่ชั่วโมง โดยใส่เป็นแบบ 24 ชั่วโมง

- Minute ให้ใส่นาที

โดยในการให้ใส่ค่าในแต่ละส่วนนี้ จะแสดงค่าของเก่าให้เห็นด้วย ถ้าต้องการใช้ค่าเก่าให้กด Enter ผ่านไป ถ้าต้องการเปลี่ยนแปลงค่าใหม่ ให้ใส่ค่าที่ต้องการลงไปให้ถูกต้อง และกด Enter ถ้ามีการใส่ไม่ถูกต้อง เครื่องจะให้ทำการใส่ค่าลงไปใหม่ และจะมีการเซ็ทเวลาใหม่ก็ต่อเมื่อมีการใส่ค่าลงไปจนถึง การใส่ค่าให้ Minute (ถ้ายกเลิกก่อน จะไม่มีการตั้งเวลาใหม่)

ในการที่จะใช้งานฟังก์ชั่นต่าง ๆ เหล่านี้ ถ้าต้องการใช้ในขณะที่ Center อยู่ในช่วงเวลาปิดเครื่องอยู่ ให้ใช้วิธีการกดคีย์ Enter + Nunber1 ค้างไว้ และทำการกดปุ่ม Reset จากนั้นก็สามารถทำการเข้าไปใช้ฟังก์ชั่นต่าง ๆ ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. การใช้เครื่อง Terminal

การใช้เครื่อง Terminal ก็คือการจะทำการรูดบัตรเพื่อที่จะผ่านเข้าออก โดยมีการใช้ดังนี้

- ทำการรูดบัตรที่เครื่องรูด โดยในการรูด จะต้องพยายามทำให้บัตรไม่เกิดการยกขึ้นหรือลอย ตลอดการรูด ถ้าการอ่านบัตรเป็นผลสำเร็จ จะมีเสียงแสดงการอ่านบัตรสำเร็จออกมา 1 ครั้ง
- เครื่องจะถามว่า จะเข้าหรือ ออก (IN or OUT) ให้กดคีย์ IN , OUT ตามต้องการ
- ถ้าในขณะนั้นต้องมีการกดรหัส เครื่องจะแสดงข้อความให้กดรหัส (Enter Code) ให้ทำการใส่รหัสลงไปจนครบ 4 ตัว (โดยเครื่องจะแสดงเป็นตัว X)
- จากนั้น Terminal จะติดต่อไปยัง Center เพื่อให้ตรวจสอบการเข้าออก และส่งผลที่ได้กลับมา และจะแสดงผลให้ทราบ ถ้าผ่านได้ ประตูก็จะเปิดพร้อมกับมีเสียงเพลงดังขึ้น ซึ่งจะต้องเข้าไปภายในเวลาประมาณ 15 วินาที

ถ้า Terminal สร้างสัญญาณเตือนออกมา แสดงว่าประตูประจำ Terminal นั้นยังไม่ได้ปิด ให้ทำการปิดประตู เสียงเตือนก็จะหายไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทสรุป และข้อเสนอแนะ

จากการวิจัยและสร้างเครื่องนี้ ได้ทำการศึกษาและออกแบบระบบต่าง ๆ ดังนี้คือ

1. โครงสร้างของระบบ

ในระบบนี้จะใช้ศูนย์กลางของระบบคือ Center เป็นที่เก็บข้อมูลของระบบ และติดต่อกับ Terminal ในการส่งข้อมูลไปให้ Terminal ทุกตัว ซึ่งมีจำนวน 16 ตัว (และเพิ่มได้สูงสุด 32 ตัว) ซึ่งนับว่ายังเป็นระบบที่ไม่ใหญ่มากนัก ซึ่งจะใช้กับระบบหรือโรงงานใหญ่ ๆ ไม่ได้ ดังนั้นในการจะพัฒนาระบบต่อไป น่าจะมีการพัฒนาระบบนี้ให้ใหญ่ขึ้น โดยอาจจะใช้ตัว Center มีจำนวนหลายตัว จากนั้นก็จะให้ใช้คอมพิวเตอร์เป็นศูนย์กลางของระบบทั้งหมด และติดต่อข้อมูลกับ Center ทั้งหมด ซึ่งจะทำให้ระบบนี้ใหญ่ขึ้นและมีประสิทธิภาพที่สูงขึ้นได้

2. ส่วนของบัตรแม่เหล็ก

ในขั้นแรกจะต้องมีการศึกษาถึงมาตรฐานของบัตรก่อน ซึ่งจะมีลักษณะการเก็บข้อมูลแบบใด จากนั้นก็จะต้องมีการศึกษาถึงวิธีการที่จะทำการอ่านข้อมูลจากบัตรออกมา ซึ่งในส่วนนี้ค่อนข้างจะมีความยากพอสมควร เนื่องจากจะต้องสร้างเครื่องอ่านบัตรขึ้นเอง และยังต้องสร้างวงจรที่ใช้ในการขยายสัญญาณจากหัวอ่าน ตลอดจนวิธีการที่จะนำสัญญาณที่ได้จากการขยาย มาทำการถอดรหัส ซึ่งจะแยกการอธิบายได้เป็นส่วน ๆ คือ

2.1 เครื่องอ่านบัตร

เครื่องอ่านบัตรที่จะต้องสร้างขึ้นนี้ จะต้องสร้างให้สามารถอ่านบัตรได้ อย่างมีประสิทธิภาพพอสมควร คือต้องการให้ในการอ่านบัตรในแต่ละครั้งจะต้องประสบผลสำเร็จในการถอดรหัสได้ทุกครั้ง หรือให้มีการผิดพลาดได้น้อยที่สุด โดยในครั้งแรกของการสร้าง ต้องการให้เครื่องอ่านบัตรนี้มีความเร็วในการอ่านบัตรเท่ากันตลอดช่วงของการอ่านบัตร ดังนั้นจึงทำการออกแบบและสร้างเครื่องอ่านแบบที่ใช้ Motor เป็นตัวควบคุมการเลื่อนบัตรในการอ่าน (ใช้บัตรเลื่อนผ่านหัวเทปที่อยู่กับที่ในการอ่านบัตร) ซึ่งก็คือเครื่องอ่านบัตรแบบอัตโนมัติ ที่แสดงอยู่ในบทที่ 2 ซึ่งในการสร้างเครื่องนี้มีความยากพอสมควร เพราะจะต้องทำให้ในขณะที่การอ่านบัตร จะต้องทำให้หัวอ่าน อ่านตรงกับแทรกที่ต้องการ ตลอดช่วงของการอ่าน จะเกิดการล้ำเข้าไปที่แทรกอื่นไม่ได้ ซึ่งจะใช้วิธีทำให้ช่องที่บัตรจะเลื่อนเข้าไปในการอ่านมีความพอดีกับบัตรมากที่สุด แต่ก็เกิดปัญหาที่ขนาดของบัตรแต่ละใบจะไม่เท่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กันทุกใบ ซึ่งจะทำให้ในบางบัตรเล็กกว่าช่องอ่านบัตร ดังนั้นจึงต้องสร้างให้มีส่วนที่เป็นสปริง ทำการดันบัตรให้ชิดไปที่ขอบด้านหนึ่ง เพื่อเป็นการแก้ปัญหานี้

และในส่วนของหัวอ่านเนื่องจากใช้หัวเทปคาสเซ็ททั่วไป ซึ่งถ้านำมาใช้ในการอ่านบัตรซึ่งเป็นบัตรแข็งแล้ว จะต้องทำให้ในเวลาการอ่าน ตัวบัตรและหัวเทป จะต้องแนบสนิทกันตลอดช่วงของการอ่าน ดังนั้นจึงต้องมีการออกแบบให้มีลักษณะที่พิเศษออกไป ซึ่งมีลักษณะที่แสดงอยู่ในบทที่ 2

เนื่องจากในการสร้างเครื่องอ่านบัตรแบบอัตโนมัตินี้ จะใช้เวลาในการสร้างค่อนข้างมาก (ประมาณ 1 เดือน ในครั้งแรกที่ทำการสร้าง ถ้าทำเครื่องต่อไปโดยใช้เครื่องเดิมเป็นต้นแบบ ก็จะใช้เวลาประมาณ 1 อาทิตย์) ดังนั้นจึงคิดเครื่องอ่านบัตรแบบใช้มือรูดขึ้นมา เพราะเห็นว่าน่าจะใช้เวลาในการสร้างน้อยกว่า (ในการสร้างเครื่องนี้ในครั้งแรกใช้เวลาประมาณ 3 วัน และถ้าสร้างเครื่องต่อไปจะใช้เวลาประมาณ 2 วัน) แต่จะต้องมีการแก้ไขโปรแกรมที่ใช้ในการถอดรหัส เพื่อให้สามารถถอดรหัสของบัตร ซึ่งใช้วิธีการรูดบัตรได้ด้วย ซึ่งเครื่องอ่านบัตรแบบใช้มือรูดนี้ จะสร้างเพียงชุดของหัวเทปและรางสำหรับการรูดบัตรเท่านั้น

ส่วนที่ยากอีกส่วนหนึ่ง ก็คือการตั้งแทรกของการอ่านของหัวอ่าน ให้ตรงกับแทรกที่ 2 ที่ต้องการอ่านให้ตรงที่สุด เพื่อที่จะได้สัญญาณในการอ่านดีที่สุด

2.2 การสร้างวงจรที่ใช้ในการขยายสัญญาณจากหัวเทป และสัญญาณที่จะใช้ในการถอดรหัส

ในการสร้างวงจรขยายจากหัวเทปจะต้องใช้ Storage Osciloscope ในการบันทึกสัญญาณในการอ่านที่ผ่านวงจรขยายแล้ว โดยจะต้องทำการปรับค่าต่าง ๆ ของวงจร เพื่อที่จะได้รูปร่างของสัญญาณที่ถูกต้องที่สุด จากนั้นก็จะต้องทำการปรับแต่งในส่วนของวงจรที่จะสร้างสัญญาณในการถอดรหัส ซึ่งจะใช้วงจรเปรียบเทียบแรงดัน ซึ่งแสดงไว้ในบทที่ 2 โดยจะทำการปรับระดับของแรงดัน reference ที่เป็นจุดเปรียบเทียบให้มีค่าเหมาะสมที่สุด เพื่อให้สามารถอ่านบัตรที่มีความแรงของสัญญาณแม่เหล็กในบัตรที่แตกต่างกันได้ทุกใบ

2.3 การออกแบบวิธีการที่จะใช้ในการถอดรหัส

ในการถอดรหัสของบัตรจะใช้โปรแกรมจาก CPU มาใช้ในการนับค่าเวลาของสัญญาณที่ใช้ในการถอดรหัส และทำการถอดรหัสจากค่านับนี้ออกมา เหตุที่ไม่ได้ใช้วงจรนับซึ่งเป็นฮาร์ดแวร์จากภายนอก ก็เนื่องจากจะทำให้สิ้นเปลืองฮาร์ดแวร์ส่วนนี้เพิ่มไปอีก และในการใช้วงจรนับจากภายนอก ซึ่งจะต้องใช้วงจรทาง Digital ในการนับ ซึ่งจะมีปัญหาที่วงจร Digital นี้จะทำงานได้ที่ความถี่สูงมาก เมื่อเกิดสัญญาณรบกวนขึ้นในระบบ ก็จะทำให้การนับค่าผิดพลาดได้ ซึ่งถ้าใช้ CPU ในการนับค่าพัลส์ เมื่อเกิดสัญญาณรบกวนความถี่สูงขึ้นในระบบซึ่ง CPU จะทำการตรวจสอบสัญญาณ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

1. ข้อมูลมาตรฐานของบัตรเอทีเอ็ม International Standard ISO 7811 / 2 1985 (E)
2. ยืน ภู่วรวรรณ, "ทฤษฎีและการประยุกต์ ไมโครโปรเซสเซอร์ Z-80" , บริษัท ซีเอ็ดยูเคชั่น จำกัด, พ.ศ. 2532
3. "คู่มือไอซี ไมโครโปรเซสเซอร์ Z80180 Z180 MPU" , บริษัท อีทีที จำกัด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. ส่วนของโปรแกรมในคอมพิวเตอร์ ที่ใช้เก็บข้อมูลเวลาเข้าออก และคำนวณเงินเดือนออกมา

โดยคอมพิวเตอร์จะเป็นตัวเก็บข้อมูลเวลาเข้าออก ของพนักงานเอาไว้ทั้งหมด โดยจะเก็บค่าทุก ๆ วันที่มีการทำงาน เมื่อถึงกำหนดเวลาที่เงินเดือนของพนักงานออก ก็จะทำการคำนวณเงินเดือนออกมา ซึ่งโปรแกรมในส่วนนี้ ใช้ภาษา C ในการเขียน ทั้งนี้เพื่อต้องการศึกษาถึงวิธีการใช้ฐานข้อมูลของภาษา C ซึ่งจะเป็นส่วนพื้นฐานของภาษา C ดังนั้นในการพัฒนาระบบต่อไปจึงแนะนำให้ใช้โปรแกรมที่ทำงานเป็นฐานข้อมูลโดยตรง เพื่อที่จะสามารถแบ่งข้อมูลของพนักงาน ออกเป็นส่วน ๆ ตามที่ต้องการได้ และสามารถนำข้อมูลไปใช้ได้กับโปรแกรมอื่น ทั่ว ๆ ไปได้ และในส่วนของการรับข้อมูลจาก Center จะใช้การส่งข้อมูลในขณะที่รันโปรแกรม Autoexec.bat ซึ่งเป็นขณะที่เปิดเครื่องคอมพิวเตอร์ ซึ่งในการพัฒนาต่อไปน่าจะเขียนโปรแกรมให้ทำการรอรับการส่งข้อมูลจาก Center โดยอาจใช้เป็น Driver ตัวหนึ่งที่อยู่ใน Memory ซึ่งจะคอยรอรับการ Interupt จากข้อมูลของ Center ที่ส่งมาให้ ซึ่งจะทำให้สามารถรับข้อมูลจาก Center ได้ตลอดเวลา แม้ในเวลาที่กำลังใช้คอมพิวเตอร์ทำงานอื่นอยู่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

1. ข้อมูลมาตรฐานของบัตรเอทีเอ็ม International Standard ISO 7811 / 2 1985 (E)
2. ยืน ภู่วรวรรณ, "ทฤษฎีและการประยุกต์ ไมโครโปรเซสเซอร์ Z-80" , บริษัท ซีเอ็ดยูเคชั่น จำกัด, พ.ศ. 2532
3. "คู่มือไอซี ไมโครโปรเซสเซอร์ Z80180 Z180 MPU" , บริษัท อีทีที จำกัด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก

โปรแกรมของเครื่อง

1. แสดงโปรแกรม Monitor ของเครื่อง Terminal และ Center ซึ่งใช้ในการทำงานของเครื่อง โดยจะใช้ Assembler ของ Z80 ในการแปลเป็นภาษาเครื่อง

2. แสดงโปรแกรมของในส่วนคอมพิวเตอร์ ที่ใช้ในการรับข้อมูลจาก Center ที่ส่งมาให้คอมพิวเตอร์ โดยใช้ภาษา Assembly ของ 8088 ร่วมกับ Function Call ของ DOS ในการเขียน

3. แสดงโปรแกรมในส่วนของคอมพิวเตอร์ที่ใช้ในการคำนวณเงินเดือนของพนักงานโดยจะใช้ภาษา C ในการเขียนโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CENTER PROGRAM

```
                ORG 0000
START:          LD B,0
STRT:           DJNZ STRT
INIT:           LD A,00
                OUTO (OMCR),A
                LD A,98H
                OUTO (CBAR),A
                LD A,8
                OUTO (BBR),A
                OUTO (CBR),A
                LD SP,8FFFH
                LD A,00
                OUTO (RCR),A
                DI
                CALL ASCSET
                CALL SETPRT
                CALL SET8255
                CALL INIT_LCD
                JP RESET

                ORG 0038H
                IN A,(CREG_D)
                RES 2,A
                OUT (CREG_D),A
                RETI

                ORG 0080H
TIMEL           EQU 20H
TIMEH           EQU 00
PORT_CONT       EQU 83H
ASCSET:         LD A,66H
                OUTO (CNTLA0),A
                OUTO (CNTLA1),A
                LD A,02
                OUTO (CNTLB0),A
                OUTO (CNTLB1),A
                LD A,4
                OUTO (STAT1),A
                RET
SETPRT:         LD A,TIMEL
                OUTO (TMDR1L),A
                OUTO (RLDR1L),A
                LD A,TIMEH
                OUTO (TMDR1H),A
                OUTO (RLDR1H),A
                LD A,4
                OUTO (TCR),A
                RET
SET8255:        LD A,8AH
                OUT (PORT_CONT),A
                LD A,0
                OUT (PORT_OUT),A
                RET
RESET:          LD A,01
                OUT (CREG_F),A
                LD A,04
                OUT (CREG_F),A
                LD A,0FH
                OUT (CREG_E),A
                XOR A
                OUT (CREG_D),A
                LD A,0EH
                OUT (PORT_KEY),A
                IN A,(PORT_KEY)
                BIT 7,A
                JR NZ, RES1
                LD A,07
                OUT (PORT_KEY),A
                IN A,(PORT_KEY)
                BIT 4,A
                JR NZ, RES2
                LD A,0FFH
                LD (BYTE_RELS_KEY),A
                CALL FUNC_SEL
                JR RES1
RES2:           BIT 5,A
                JR NZ, RES3
                CALL SET_MEM_CARD
                JR RES1
RES3:           BIT 6,A
                JR NZ, RES4
                XOR A
                LD (BYTE_RETRY_SEND),A
                JP POWER_ON
RES4:           LD A,0BH
                OUT (PORT_KEY),A
                IN A,(PORT_KEY)
                BIT 4,A
                JR NZ, RES1
                CALL SET_MEM_CARD
                CALL LD_CARD_DATA
RES1:           LD IX,CENTER_CFG+1
                CALL CHK_CFG_TIME
                OR A
                JP NZ, SYSTEM_STOP
                LD IX,CENTER_CFG+3
                CALL CHK_CFG_TIME
                OR A
                JP Z, SYSTEM_STOP
                JP MAIN

SYSTEM_STOP:    LD A,0EH
                OUT (CREG_E),A
                CALL CLS_LCD
                LD A,0BH
                OUT (P_COMND),A
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                EI
                LD A,20H
                OUTO (ICR),A
                SLP
INT:            LD IX,CENTER_CFG+1
                CALL CHK_CFG_TIME
                OR A
                JP NZ, INT2
                LD IX,CENTER_CFG+3
                CALL CHK_CFG_TIME
                OR A
                JP Z, INT2
                JP POWER_ON
INT2:           LD A,(BYTE_RETRY_SEND)
                BIT 7,A
                JR Z, SYSTEM_STOP
                AND A,1FH
                JR NZ, INT1
                LD A,20H
                LD (BYTE_RETRY_SEND),A
                JR SYSTEM_STOP
INT1:           XOR A
                OUTO (ICR),A
                LD A,0FH
                OUT (P_COMND),A
                CALL CLS_LCD
                CALL COMPUTER
                JR SYSTEM_STOP
POWER_ON:       LD A,0FH
                OUT (CREG_E),A
                XOR A
                OUTO (ICR),A
                LD A,0FH
                OUT (P_COMND),A
                CALL CLS_LCD
                LD HL,BYTE_RETRY_SEND
                LD A,RETRY_SEND
                OR A,0C0H
                BIT 7,(HL)
                JR Z, 02
                SET 5,A
                LD (HL),A
                CALL DEL_WORK_TIME
                JP MAIN

BYTE_OLD_TIME   EQU 840CH
RETRY_SEND      EQU 3
MAIN:           CALL DISPLAY_TIME
MAINL:          LD HL,(BYTE_OLD_TIME)
                CALL INPUT_TIME
                XOR A
                SBC HL,DE
                JR Z, MAIN1
                LD (BYTE_OLD_TIME),DE
                CALL DISPLAY_TIME
                LD IX,CENTER_CFG+5
                CALL CHK_CFG_TIME
                OR A
                JR NZ, TP1
                LD A,(BYTE_RETRY_SEND)
                BIT 7,A
                JR Z, TP1
                BIT 6,A
                JR Z, TP1
                CALL COMPUTER
TP1:            LD IX,CENTER_CFG+3
                CALL CHK_CFG_TIME
                OR A
                JR NZ, MAIN1
                JP SYSTEM_STOP
MAIN1:          CALL TERMINAL
                CALL KEY_PRO
                JR MAINL

;       ** TERMINAL PROGRAM **
                ORG 0300H
DATA_SEND_BUF   EQU 8000H
DATA_RECEIVE_BUF EQU 8100H
TEMP            EQU 8200H
BYTE_TERMINAL   EQU 8400H
CENTER_CFG      EQU 9000H
TERMINAL_CFG    EQU 9100H
CARD_DATA_EXT   EQU 9200H
MODE_RQS        EQU 0D1H
CARD_RQS        EQU 0D3H
SYNC            EQU 16H
STX             EQU 02H
ETX             EQU 03H
EOT             EQU 04H
PORT_OUT        EQU 80H
TERMINAL:       OUT (0E0H),A
                LD HL,DATA_SEND_BUF
                LD A,(BYTE_TERMINAL)
                LD (HL),A
                LD (HL),0D2H
                CALL CRC_WRITE
                LD (HL),EOT
                CALL RS485_SEND
                CALL RS485_RECV
                JR NZ,END_T
                LD HL,DATA_RECEIVE_BUF+1
                LD A,(HL)
                CP MODE_RQS
                JR Z,D1
                CP CARD_RQS
                JR Z,D3
                JR END_T
D1:             CALL SET_ANS_MODE
                CALL RS485_SEND
                LD A,(HL)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                CP STX
                JR NZ,END_T
                INC HL
                CALL CP_CARD_DATA
                CALL SET_ANS_CARD
                CALL RS485_SEND
END_T:          CALL TERMINAL_TIME
                LD HL,BYTE_TERMINAL
                INC (HL)
                LD A,(CENTER_CFG+0)
                CP (HL)
                LD (HL),80H
                RET

DELAY:          INO A,(TCR)
                RES 0,A
                PUSH AF
                OUTO (TCR),A
                OUTO (TMDR0L),L
                OUTO (RLDR0L),L
                OUTO (TMDR0H),H
                OUTO (RLDR0H),H
                SET 0,A
                OUTO (TCR),A
DL1:            INO A,(TCR)
                BIT 6,A
                JR Z, DL1
                INO A,(TMDR0H)
                DJNZ DL1
                POP AF
                OUTO (TCR),A
                RET

;       ** RS485 DATA SEND (To Terminal) **

RS485_SEND:     LD HL,DATA_SEND_BUF
                IN A,(PORT_OUT)
                SET 0,A
                OUT (PORT_OUT),A
                LD A,66H
                OUTO (CNTLA1),A
                LD A,SYNC
                OUTO (TDR1),A
ST1:            INO A,(STAT1)
                BIT 1,A
                JR Z, ST1
                LD A,(HL)
                OUTO (TDR1),A
                CP EOT
                JR Z, ST2
                INC L
                JR NZ, ST1
ST2:            LD HL,672
                LD B,3
                CALL DELAY
                IN A,(PORT_OUT)
                RES 0,A
                OUT (PORT_OUT),A
                LD HL,600
                LD B,1
                CALL DELAY
                RET

;       ** RS485 DATA RECEIVE (from terminal) **

DATA_TIME       EQU 500H
RS485_RECV:     LD HL,DATA_RECEIVE_BUF
                INO A,(STAT1)
                BIT 7,A
                JR Z, ERR_R
                AND A,30H
                JR NZ, ERR_R
                INO A,(RDR1)
                CP SYNC
                JR NZ, ERR_R
RT1:            LD BC,DATA_TIME
RT2:            INO A,(STAT1)
                BIT 7,A
                JR NZ, RT3
                DEC BC
                LD A,C
                OR A,B
                JR Z, ERR_R
                JR RT2
RT3:            AND A,30H
                JR NZ, ERR_R
                INO A,(RDR1)
                LD (HL),A
                CP EOT
                JR Z, RT4
                INC L
                JR Z, ERR_R
                JR RT1
RT4:            CALL CRC_CHECK
                RET
ERR_R:          LD A,0FFH
                RET

;       ** COMPARE CARD DATA & CHECK TIME **

CP_CARD_DATA:   CALL SET_CARD_DATA
                CALL T_CFG_CAL
                LD D,(IX+10)
                LD E,(IX+11)
CPA:            CALL CP_DATA
                CP 30H
                JR Z, CP1
                CP 31H
                JR Z, CP2
                CP 32H
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                JR Z, CP3
CP2:            LD A,E
                AND A,0F0H
                ADD A,10H
                JR NC,01
                INC D
                LD E,A
                LD A,(DE)
                CP 0FFH
                JR Z, CP7
                CP 0EFH
                JR Z, CP7
                JR CPA
CP1:            CALL CHECK_TIME
                RET
CP3:            LD A,32H
                RET
CP7:            LD DE,CARD_DATA_EXT
CP8:            CALL CP_DATA
                CP 30H
                JR Z, CP11
                CP 32H
                RET Z
CP9:            LD A,E
                AND A,0F0H
                ADD A,10H
                JR NC,01
                INC D
                LD E,A
                LD A,(DE)
                CP 0FFH
                JR Z,06
                CP 0EFH
                JR Z,02
                LD A,31H
                RET
CP11:           LD A,(DE)
                LD H,A
                INC DE
                LD A,(DE)
                LD L,A
                LD A,(BYTE_TERMINAL)
                SUB A,80H
                INC A
                LD B,A
CP12:           SRL H
                RR L
                DJNZ CP12
                JR NC, CP13
                LD A,30H
                RET
CP13:           LD A,31H
                RET
SET_CARD_DATA:  LD DE,TEMP
                LD B,08H
                CALL COM_DATA
                LD A,(HL)
                CP 1DH
                JR Z, SD2
                LD A,31H
                POP HL
                RET
SD2:            INC HL
                PUSH HL
                INC HL
                LD A,(HL)
                CP ETX
                JR Z, SD3
                LD B,2
                CALL COM_DATA
SD4:            POP HL
                LD A,(HL)
                LD (DE),A
                RET
SD3:            LD A,0FFH
                LD (DE),A
                INC DE
                LD (DE),A
                INC DE
                JR SD4

COM_DATA:       LD C,(HL)
                SLA C
                SLA C
                SLA C
                SLA C
                INC HL
                LD A,(HL)
                AND A,0FH
                OR A,C
                LD (DE),A
                INC DE
                INC HL
                DJNZ COM_DATA
                RET

;       ---------- COMPARE DATA ----------

CP_DATA:        PUSH DE
                LD HL,TEMP
                LD B,8
CD1:            LD A,(DE)
                CP (HL)
                JR NZ, CD2
                INC DE
                INC HL
                DJNZ CD1
                LD A,(HL)
                CP 0FFH
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                JR Z, CD3
                LD B,2
CD5:            LD A,(DE)
                CP (HL)
                JR NZ, CD4
                INC DE
                INC HL
                DJNZ CD5
CD3:            LD A,30H
                JR CD_E
CD2:            LD A,31H
                JR CD_E
CD4:            LD A,32H
CD_E:           POP DE
                LD B,0AH
CD6:            INC DE
                DJNZ CD6
                RET

T_CFG_CAL:      LD DE,TERMINAL_CFG
                LD A,(BYTE_TERMINAL)
                SLA A
                SLA A
                SLA A
                SLA A
                LD E,A
                RET

;       ------- PROGRAM CHECK TIME -------

IN              EQU 59H
OUT             EQU 4EH
CHECK_TIME:     LD A,(TEMP+10)
                CP IN
                JR NZ, CT3
                LD A,(DE)
                BIT 7,A
                JR NZ, CT1
                LD A,30H
                RET
CT1:            PUSH DE
                CALL T_CFG_CAL
                INC DE
                INC DE
                EX DE,HL
                CALL COMPARE_TIME
                CP 00
                JR NZ, CT2
                POP HL
                CALL INPUT_TIME
                LD (HL),D
                INC HL
                LD (HL),E
                LD A,33H
                RET
CT2:            POP DE
                LD A,30H
                RET
CT3:            PUSH DE
                CALL T_CFG_CAL
                LD B,6
CT4:            INC DE
                DJNZ CT4
                EX DE,HL
                CALL COMPARE_TIME
                CP 00
                JR NZ, CT5
                POP HL
                INC HL
                INC HL
                CALL INPUT_TIME
                LD (HL),D
                INC HL
                LD (HL),E
                LD A,34H
                RET
CT5:            POP DE
                LD A,35H
                RET

COMPARE_TIME:   PUSH DE
                PUSH HL
                LD D,(HL)
                INC HL
                LD E,(HL)
                LD (TEMP+80H),DE
                INC HL
                LD D,(HL)
                INC HL
                LD E,(HL)
                LD (TEMP+82H),DE
                LD HL,(TEMP+80H)
                XOR A
                SBC HL,DE
                JR NC, CPT1
                CALL INPUT_TIME
                LD HL,(TEMP+80H)
                XOR A
                SBC HL,DE
                JR NC, CPT3
                LD HL,(TEMP+82H)
                XOR A
                SBC HL,DE
                JR C, CPT3
                JR CPT4
CPT1:           CALL INPUT_TIME
                LD HL,(TEMP+82H)
                XOR A
                SBC HL,DE
                JR C, CPT3
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                LD HL,(TEMP+80H)
                XOR A
                SBC HL,DE
                JR NC, CPT3
CPT4:           LD A,00
                JR 02
CPT3:             LD A,0FFH
                POP HL
                POP DE
                RET

SET_ANS_MODE:   LD HL,DATA_SEND_BUF
                LD A,(BYTE_TERMINAL)
                LD (HL),0D6H
                INC HL
                LD (HL),STX
                INC HL
                LD (HL),0FH
                INC HL
                CALL T_CFG_CAL
                INC E
                LD A,(DE)
                LD (HL),A
                INC HL
                LD (HL),STX
                CALL CRC_WRITE
                LD (HL),EOT
                RET

SET_ANS_CARD:   LD HL,DATA_SEND_BUF
                PUSH AF
                LD A,(BYTE_TERMINAL)
                LD (HL),A
                INC HL
                LD (HL),0D8H
                INC HL
                LD (HL),STX
                INC HL
                LD (HL),10H
                INC HL
                POP AF
                LD (HL),A
                INC HL
                LD (HL),ETX
                CALL CRC_WRITE
                LD (HL),EOT
                RET

BYTE_SECOND1    EQU 8410H
ALL_ADDRESS     EQU A0H
TERMINAL_TIME:  CALL INPUT_SECOND
                LD HL,BYTE_SECOND1
                CP (HL)
                RET Z
                BIT 0,A
                RET NZ
                LD (HL),A
SET_TIME_SEND:  LD HL,DATA_SEND_BUF
                LD (HL),ALL_ADDRESS
                INC HL
                LD (HL),0D0H
                INC HL
                LD (HL),STX
                INC HL
                LD (HL),0EH
                INC HL
                CALL INPUT_TIME
                SRL A
                SRL A
                SRL A
                ADD A,30H
                LD (HL),A
                INC HL
                LD A,D
                AND A,0FH
                ADD A,30H
                LD (HL),A
                INC HL
                LD A,E
                SRL A
                SRL A
                SRL A
                SRL A
                ADD A,30H
                LD (HL),A
                INC HL
                LD A,E
                AND A,0FH
                ADD A,30H
                LD (HL),A
                INC HL
                LD (HL),ETX
                CALL CRC_WRITE
                LD (HL),EOT
                CALL RS485_SEND
                RET

;       ---- PROGRAM KEY SCAN -----
BYTE_RELS_KEY   EQU 8402H
STATUS_BYTE     EQU 8404H
BYTE_TIME_WAIT  EQU 8406H
KEY_BUFFER      EQU 8440H
SCANKEY_ADDH    EQU 1FH
WAIT_DIS        EQU 0200H
LAST_FUNC       EQU 34H
PORT_IN         EQU 81H
PORT_KEY        EQU 82H
FUNCTION        EQU 46H
ENTER           EQU 0DH
ESC             EQU 1BH
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
YES             EQU 59H
NO              EQU 4EH
LEFT            EQU 59H
RIGHT           EQU 4EH
DEL             EQU 08H
KEY_PRO:        CALL KEY_SCAN
                OR A,A
                RET NZ
                LD A,(HL)
                CP FUNCTION
                RET NZ
                PUSH HL
                LD HL,MESSAGE+2F0H
                LD A,40H
                CALL DISPLAY
                LD A,40H
                CALL GOTO
                POP HL
                LD E,(HL)
                CALL WR_BYTE
                CALL BEEP
                CALL INPUT_KEYDATA
                OR A,A
                JR NZ,03
                CALL CHK_COMMAND
                CALL DISPLAY_TIME
                RET

KEY_SCAN:       PUSH BC
                LD B,4
                LD D,0EEH
                LD C,PORT_KEY
SK1:            OUT (C),D
                IN E,(C)
                LD A,E
                AND A,0F0H
                LD E,A
                CP 0F0H
                JR NZ, SK3
                RLC D
                DJNZ SK1
                LD A,(BYTE_RELS_KEY)
                OR A,A
                JR Z, SKD
                LD HL,9000H
                LD B,01
                    CALL DELAYK
                XOR A
                LD (BYTE_RELS_KEY),A
                JR SKD
SK3:            LD A,(BYTE_RELS_KEY)
                OR A,A
                JR NZ, SKD
                XOR A
SK4:            SRL D
                JR NC,03
                INC A
                JR SK4
                SLA A
                SLA A
SK5:            SLA E
                JR NC,03
                INC A
                JR SK5
                OR A,0F0H
                LD L,A
                LD H,SCANKEY_ADDH
                LD A,0FFH
                LD (BYTE_RELS_KEY),A
                XOR A
                JR 02
SKD:            LD A,0FFH
                POP DE
                POP BC
                RET

BEEP:           PUSH HL
                LD A,6
                OUTO (TCR),A
                LD HL,8000H
                LD B,1
                CALL DELAY
                LD A,4
                OUTO (TCT),A
                POP BC
                POP HL
                RET
DELAYK:         PUSH BC
                LD B,10H
DELAY1:         CALL DELAYL
                DJNZ DELAY1
                POP BC
                RET

;       ---- PROGRAM INPUT KEY DATA ----
INPUT_KEYDATA:  PUSH BC
                CALL RD_ADDRESS
IC1:            CALL INPUT_KEY
                LD A,E
                CP 30H
                JR C, IC2
                CP 40H
                JR NC, IC2
                CALL WR_BYTE
                JR IC1
IC2:            CP DEL
                JR NZ, IC3
                CALL RD_ADDRESS
                CP C
                JR Z, IC1
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                CALL DEL_BYTE
IC3:            CP ENTER
                JR NZ, IC4
                LD A,C
                CALL READ_DATA
                XOR A
                JR IC5
IC4:            CP ESC
                JR Z, IC5
                CP 0FFH
                JR NZ, IC1
IC5:            POP BC
                RET

;       ---- PROGRAM CHECK COMMAND ----
MESSAGE         EQU 1B00H
FUNC            EQU 30H
CHK_COMMAND:    LD HL,KEY_BUFFER
                LD A,(HL)
                CP 20H
                JR Z, FUNC_SEL
                LD A,(HL)
                CP 20H
                JR NZ, CCE
                CALL CLS_LCD
                DEC HL
                LD A,(HL)
CC1:            CP FUNC+1
                JP Z, FUNCTION1
                CP FUNC+2
                JP Z, FUNCTION2
                CP FUNC+3
                JP Z, FUNCTION3
                CP FUNC+4
                JP Z, FUNCTION4
CCE:            XOR A
                LD HL,MESSAGE+0
                CALL DISPLAY
                CALL WAIT_KEY
                RET

FUNC_SEL:       CALL DIS_PRESSK
                LD B,31H
FS1:            CALL DIS_FUNC
FS2:            CALL INPUT_KEY
                CP LEFT
                JR NZ, FS3
                LD A,B
                CP 32H
                JR C, FS2
                DEC B
                JR FS1
FS3:            CP RIGHT
                JR NZ, FS4
                LD A,B
                CP LAST_FUNC
                JR NC, FS2
                JR FS1
FS4:            CP ENTER
                JR NZ, FS5
                CALL CLS_LCD
                JP CC1
FS5:            CP ESC
                JR NZ, FS6
FS7:            CALL CLS_LCD
                RET
FS6:            CP 0FFH
                JR Z, FS7
                JR FS2

MESSAGEF        EQU 1E00H
DIS_FUNC:       LD A,B
                CP 31H
                JR NZ,05
                LD HL,MESSAGEF+0
                JR DFE
                CP 32H
                JR NZ,05
                LD HL,MESSAGEF+10H
                JR DFE
                CP 33H
                JR NZ,05
                LD HL,MESSAGEF+20H
                JR DFE
                CP 34H
                JR NZ,05
                LD HL,MESSAGEF+30H
                JR DFE
                CP 35H
                JR NZ,05
                LD HL,MESSAGEF+40H
DFE:            LD A,00
                CALL DISPLAY
                RET

DIS_PRESSK:     CALL CLS_LCD
                LD HL,MESSAGE+10H
                LD A,40H
                CALL DISPLAY
                RET

;       ---- PROGRAM DISPLAY TIME ----

DISPLAY_TIME:   LD HL,MESSAGE+20H
                CALL CLS_LCD
                LD A,00H
                CALL DISPLAY
                LD A,05
                CALL GOTO
                CALL INPUT_TIME
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                EX DE,HL
                LD A,H
                CALL SHIFTR
                LD A,H
                AND A,0FH
                ADD A,30H
                LD E,A
                CALL WR_BYTE
                LD E,3AH
                CALL WR_BYTE
                LD A,L
                CALL SHIFTR
                LD A,L
                AND A,0FH
                ADD A,30H
                LD E,A
                CALL WR_BYTE
                LD A,40H
                CALL GOTO
                LD A,(BYTE_RETRY_SEND)
                BIT 5,A
                RET Z
                LD HL,MESSAGE+270H
                LD A,40H
                CALL DISPLAY
                RET
SHIFTR:         SRL A
                SRL A
                SRL A
                SRL A
                ADD A,30H
                LD E,A
                CALL WR_BYTE
                RET

;       ---- PROGRAM INPUT KEY ----

WAITK_TIME      EQU 2000H
INPUT_KEY:      PUSH IX
                PUSH HL
                LD HL,WAITK_TIME
                LD (TEMP+1E01H),HL
IK1:            CALL KEY_SCAN
                OR A,A
                JR Z, IK2
                CALL TERMINAL
                LD HL,(TEMP+1E0H)
                DEC HL
                LD (TEMP+1E0H),HL
                LD A,L
                OR A,H
                JR NZ, IK1
                LD E,0FFH
                JR IKD
IK2:            LD E,(HL)
                CALL BEEP
IKD:            POP HL
                POP IX
                RET

;       ---- FUNCTION PROGRAM ----

;       ---- FUNCTION 1 ----
;       (Edit Terminal & Center Config)
FUNCTION1:      XOR A
F11:            CALL CLS_LCD
                LD HL,MESSAGE+30H
                CALL FUNCP
                CP ENTER
                JR Z, FN12
                CP RIGHT
                JR Z, F11
                CP ESC
                RET Z
                CP 0FFH
                RET Z
F12:            CALL CLS_LCD
                LD HL,MESSAGE+40H
                CALL FUNCP
                CP ENTER
                JP Z, FN12
                CP LEFT
                JR Z, F11
                CP ESC
                RET Z
                CP 0FFH
                RET Z
                JR F12

FN11:           LD IX,CENTER_CFG+2
                LD B,05
                LD IY,MESSAGE+50H
FN111:          CALL CLS_LCD
                PUSH IY
                POP HL
                XOR A
                CALL DISPLAY
                PUSH IX
                POP HL
                CALL DIS_OLD_TIME
                CALL INPUT_KEYDATA
                JR Z, FN112
                CP ESC
                JR Z, F11
                RET
FN112:          CALL CHK_INPUT_TIME
                CP 00
                JR Z,09
                CP 0DDH
                JR Z, FN113
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        JR FN113
        LD HL,(KEY_BUFFER+10)
        LD (IX+0),L
        LD (IX+1),H
FN113:  INC IX
        INC IX
        LD DE,10H
        ADD IY,DE
        DJNZ FN111
        JP F12

FN12:   CALL CLS_LCD
        LD HL,MESSAGE+80H
        XOR A
        CALL DISPLAY
        LD A,40H
        CALL GOTO
        CALL INPUT_KEYDATA
        OR A,A
        JR Z, FN121
        CP ESC
        JP Z, F12
        RET
FN121:  CALL CHK_INPUT_TNL
        CP 00
        JR NZ, FN12
FN127:    CALL CLS_LCD
        LD HL,MESSAGE+90H
        XOR A
        CALL DISPLAY
        CALL DIS_OLD_MODE
        CALL INPUT_KEYDATA
        JR Z, FN122
        CP ESC
        JR Z, FN12
        RET
FN122:  LD HL,KEY_BUFFER
        LD A,(HL)
        CP 20H
        JR Z, FN123
        INC HL
        LD A,(HL)
        CP 2011
        JR NZ, FN127
        DEC HL
        LD A,(HL)
        CP 34H
        JR NC, FN127
        LD A,(KEY_BUFFER)
        LD (IX+1),A
FN123:  INC IX
        LD B,04
        LD IY,MESSAGE+A0H
FN126:  CALL CLS_LCD
        PUSH IY
        POP HL
        XOR A
        CALL DISPLAY
        PUSH IX
        POP HL
        CALL DIS_OLD_TIME
        CALL INPUT_KEYDATA
        OR A,A
        JR Z, FN124
        CP ESC
        JR Z, FN12
        RET
FN124:  CALL CHK_INPUT_TIME
        JR Z,06
        CP 0DDH
        JR Z, FN125
        JR FN126
        LD HL,(KEY_BUFFER+10)
        LD (IX+0),L
        LD (IX+1),H
FN125:  INC IX
        INC IX
        LD DE,10H
        ADD IY,DE
        DJNZ FN126
        JP FN12

FUNCP:  XOR A
        CALL DISPLAY
        CALL INPUT_KEY
        RET

DIS_OLD_MODE:  LD A,40H
        CALL GOTO
        LD E,28H
        CALL WR_BYTE
        LD E,(IX+1)
        CALL WR_BYTE
        LD E,29H
        CALL WR_BYTE
        LD E,3AH
        CALL WR_BYTE
        RET

CHK_INPUT_TIME:  PUSH BC
        PUSH IX
        LD HL,KEY_BUFFER
        LD DE,KEY_BUFFER+10
        LD A,(HL)
        CP 20H
        JR Z, CIT2
        PUSH HL
        POP IX
        LD A,(IX+4)
        CP 20H
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                JR NZ, CITE
                LD A,(IX+3)
                CP 20H
                JR Z, CITE
                CALL SHIFTL
                CP 24H
                JR NC, CITE
                INC DE
                CALL SHIFTL
                CP 60H
                JR NC, CITE
                LD A,00
                JR CITD
CIT2:           LD A,0DDH
                JR CITD

CITE:           LD A,0FFH
CITD:           POP IX
                POP BC
                RET
SHIFTL:         LD B,(HL)
                SLA B
                SLA B
                SLA B
                SLA B
                INC HL
                LD A,(HL)
                AND A,0FH
                OR A,B
                LD (DE),A
                RET

CHK_INPUT_TNL:  LD IX,KEY_BUFFER
                LD DE,TERMINAL_CFG
                LD A,(IX+0)
                CP 20H
                JR Z, CKTE
                LD A,(IX+1)
                CP 20H
                JR Z, CKT1
                LD A,(IX+2)
                CP 20H
                JR NZ, CKTE
                LD A,(IX+0)
                CP 32H
                JR NC, CKTE
                CP 31H
                JR NZ, CKT2
                LD E,10
CKT2:           LD A,(IX+1)
CKT3:           AND A,0FH
                ADD A,E
                CP 00
                JR Z, CKTE
                CP 17
                JR NC, CKTE
                LD E,A
                DEC E
                SLA E
                SLA E
                SLA E
                SLA E
                PUSH DE
                POP IX
                LD A,00
                RET
CKT1:           LD A,(IX+0)
                JR CKT3
CKTE:           LD A,0FFH
                RET

DIS_OLD_TIME:   LD A,40H
                CALL GOTO
                LD E,28H
                CALL WR_BYTE
                LD A,(HL)
                CALL SHIFTR
                LD A,(HL)
                AND A,0FH
                ADD A,30H
                CALL WR_BYTE
                LD E,3AH
                CALL WR_BYTE
                INC HL
                LD A,(HL)
                CALL SHIFTR
                LD A,(HL)
                AND A,0FH
                ADD A,30H
                CALL WR_BYTE
                LD E,29H
                CALL WR_BYTE
                LD E,3AH
                CALL WR_BYTE
                RET

;       -- FUNCTION 2 --
;     -- ADD CARD OR  EDIT CARD --

WAIT_CARD_TIME  EQU 2000H
FUNCTION2:      CALL CLS_LCD
                LD HL,WAIT_CARD_TIME
                LD (BYTE_TIME_WAIT),HL
                IN A,(PORT_IN)
                BIT 2,A
                JR Z,05
                CALL CARD_OUT
                JR F21
                CALL DIS_INS_CARD
F21:            CALL CARD_CHECK
                OR A,A
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                JR Z, F22
                CP 0EEH
                JR Z, F23
                LD HL,MESSAGE+F0H
                XOR A
                CALL DISPLAY
                CALL WAIT_KEY
                CP 0FFH
                RET Z
                JR FUNCTION2
F23:            CALL KEY_SCAN
                JR NZ,04
                LD A,(HL)
                CP ESC
                RET Z
                CALL TERMINAL
                LD HL,(BYTE_TIME_WAIT)
                DEC HL
                LD (BYTE_TIME_WAIT),HL
                OR A,H
                RET Z
                JR F21
F22:            LD HL,MESSAGE+110H
                XOR A
                CALL DISPLAY
F24:            LD HL,MESSAGE+120H
                LD A,40H
                CALL DISPLAY
F26:            CALL INPUT_KEY
                LD A,E
                CP ENTER
                JP Z, FN21
                CP ESC
                JP Z, CARD_OUT
                CP 0FFH
                JP Z, CARD_OUT
                CP RIGHT
                JR Z, F25
                JR F26
F25:            LD HL,MESSAGE+130H
                LD A,40H
                CALL DISPLAY
F27:             CALL INPUT_KEY
                LD A,E
                CP ENTER
                JP Z, FN221
                CP ESC
                JP Z, CARD_OUT
                CP 0FFH
                JP Z, CARD_OUT
                CP LEFT
                JR Z, F24
                JR F27
DIS_INS_CARD:   LD HL,MESSAGE+E0H
                XOR A
                CALL DISPLAY
                RET

FN21:           CALL CLS_LCD
                LD HL,MESSAGE+80H
                XOR A
                CALL DISPLAY
                LD A,40H
                CALL GOTO
                CALL INPUT_KEYDATA
                CP 00
                JR Z, FN22
                CP ESC
                JR Z,03
                JP CARD_OUT
                CALL CARD_OUT
FN22:           CALL CHK_INPUT_TNL
                JR NZ, FN21
                CALL CLS_LCD
                CALL CHK_CARD_NORM
                CP 0FFH
                JR NZ, FN24
                LD HL,MESSAGE+160H
                XOR A
                CALL DISPLAY
                CALL WAIT_KEY
                CP 0FFH
                JP Z, CARD_OUT
                CALL CARD_OUT
                JP F21
FN24:           CP 00
                JR NZ,05
                LD HL,MESSAGE+140H
                JR 03
                LD HL,MESSAGE+150H
                XOR A
                CALL DISPLAY
                CALL WAIT_KEY
                CP 0FFH
                JP Z, CARD_OUT
FN25:           LD B,2
                LD IX,TEMP+188H
                LD IY,MESSAGE+170H
FN26:           CALL CLS_LCD
                XOR A
                CALL DISPLAY
                LD A,40H
                CALL GOTO
                CALL INPUT_KEYDATA
                CP 00
                JR Z, FN27
                CP ESC
                JP Z, FN21
                JP CARD_OUT
FN27:           CALL CHK_INPUT_CODE
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        CP 00
        JR NZ, FN26
        LD HL,(KEY_BUFFER+10)
        LD (IX+0),L
        LD (IX+1),H
        INC IX
        LD DE,10H
        ADD IY,DE
        DJNZ FN26
        LD DE,(TEMP+1F0H)
        LD HL,TEMP+180H
        LD BC,10
        LDIR
        LD B,4
        EX DE,HL
FN28:   LD (HL),0FFH
        INC HL
        DJNZ FN28
        EX DE,HL
        LD BC,2
        LDIR
        CALL CARD_OUT

FN221:  CALL CLS_LCD
        CALL CHK_CARD_EXT
        CP 0FFH
        JR NZ, FN222
        LD HL,MESSAGE+160H
        XOR A
        CALL DISPLAY
        CALL WAIT_KEY
        JP CARD_OUT
FN222:  CP 00
        JR NZ,05
        LD HL,MESSAGE+140H
        JR 03
        LD HL,MESSAGE+150H
        XOR A
        CALL DISPLAY
        CALL WAIT_KEY
        CP 0FFH
        JP Z, CARD_OUT
FN223:  LD B,2
        LD IX,TEMP+100H
        LD IY,MESSAGE+170H
FN224:  CALL CLS_LCD
        XOR A
        CALL DISPLAY
        LD A,40H
        CALL GOTO
        CALL INPUT_KEYDATA
        CP 00
        JR Z, FN225
        CP ESC
        JP Z, F22
        JP CARD_OUT
FN225:  CALL CHK_INPUT_CODE
        CP 00
        JR NZ, FN224
        LD HL,(KEY_BUFFER+10)
        LD (IX+0),L
        LD (IX+1),H
        INC IX
        LD DE,10H
        ADD IY,DE
        DJNZ FN224
        CALL CLS_LCD
        LD HL,MESSAGE+190H
        XOR A
        CALL DISPLAY
        LD HL,MESSAGE+1A0H
        LD A,40H
        CALL DISPLAY
        LD BC,1030H
        LD HL,0
        LD D,31H
FN226:  LD A,09
        CALL GOTO
        LD E,C
        CALL WR_BYTE
        LD E,D
        CALL WR_BYTE
FN227:  PUSH DE
        CALL INPUT_KEY
        LD A,E
        CP ESC
        JP Z, F22
        CP 0FFH
        JP Z, CARD_OUT
        CP YES
        JR NZ,03
        SCF
        JR 05
        CP NO
        JR NZ, FN227
        XOR A
        INC D
        LD A,D
        CP 3AH
        JR NZ,03
        LD D,30H
        INC C
        DJNZ FN226
        LD IX,(TEMP+18AH)
        LD (TEMP+18CH),IX
        LD (TEMP+18AH),HL
        LD HL,TEMP+180H
        LD DE,(TEMP+1F0H)
        LD BC,12
        LDIR
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                EX DE,HL
                LD (HL),0
                INC HL
                LD (HL),0
                INC HL
                LD BC,2
                EX DE,HL
                LDIR
                CALL CARD_OUT
                JP F21

CARD_CHECK:     IN A,(PORT_IN)
                BIT 2,A
                JR NZ, AC2
                LD HL,STATUS_BYTE
                BIT 5,(HL)
                JR Z, AC1
                PUSH HL
                CALL DIS_INS_CARD
                POP HL
                RES 0,(HL)
AC1:            LD A,0EEH
                RET
AC2:            BIT 3,A
                JR NZ, AC1
                LD HL,STATUS_BYTE
                BIT 0,(HL)
                JR NZ, AC1
                SET 0,(HL)
                IN A,(PORT_OUT)
                SET 6,A
                OUT (PORT_OUT),A
                RES 6,A
                OUT (PORT_OUT),A
                CALL CARD_COUNT
                JR NZ, CARD_ERR
                CALL PROG_12
                OR A,A
                JR NZ, CARD_ERR
                CALL CARD_CRC_CHK
                OR A,A
                JR NZ, CARD_ERR
                PUSH BC
                LD HL,TEMP+1
                LD DE,TEMP+180H
                LD B,8
                CALL COM_CARD_DATA
                POP BC
                XOR A
                RET
CARD_ERR:       LD A,0FFH
                RET
COM_CARD_DATA:  LD C,(HL)
                SLA C
                SLA C
                INC HL
                LD A,(HL)
                AND A,0FH
                OR A,C
                LD (DE),A
                INC DE
                INC HL
                INC HL
                DJNZ COM_CARD_DATA
                RET

CARD_OUT:       LD HL,STATUS_BYTE
                SET 0,(HL)
                CALL CLS_LCD
                LD HL,MESSAGE+1B0H
                XOR A
                CALL DISPLAY
                IN A,(PORT_OUT)
                SET 7,A
                OUT (PORT_OUT),A
                RES 7,A
                OUT (PORT_OUT),A
                RET

WAIT_KEY:       LD HL,MESSAGE+100H
                LD A,40H
                CALL DISPLAY
                CALL INPUT_KEY
                LD A,E
                RET

CHK_INPUT_CODE: PUSH BC
                PUSH IX
                LD HL,KEY_BUFFER
                LD DE,KEY_BUFFER+10
                LD A,(HL)
                CP 20H
                JR Z, CICE
                PUSH HL
                POP IX
                LD A,(IX+4)
                CP 20H
                JR NZ, CICE
                LD A,(IX+3)
                CP 20H
                JR Z, CICE
                CALL SHIFTL
                INC HL
                INC DE
                CALL SHIFTL
                XOR A
                JR CICD
CICE:           LD A,0FFH
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
CICD:           POP IX
                POP BC
                RET

CHK_CARD_EXT:   PUSH HL
                PUSH BC
                LD DE,CARD_DATA_EXT
                JP CN0

CHK_CARD_NORM:  PUSH HL
                PUSH DE
                PUSH BC
                LD D,(IX+10)
                LD E,(IX+11)
CN0:            LD HL,TEMP+180H
                LD B,8
CN1:            LD A,(DE)
                CP 0FFH
                JR Z, CN3
                CP 0EFH
                JR Z, CN4
                CP (HL)
                JR NZ, CN2
                INC DE
                DJNZ CN1
                LD A,E
                AND A,0F0H
                LD E,A
                LD (TEMP+1F0H),DE
                LD A,11H
                JR CND
CN2:            LD A,E
                AND A,0F0H
                ADD A,10H
                JR NC,01
                LD E,A
                JR CN0
CN3:            LD (TEMP+1F0H),DE
                XOR A
                JR CND
CN4:            LD A,0FFH
CND:            POP BC
                POP HL
                RET

;       ------ FUNCTION 3 (Delete card) ------

FUNCTION3:      XOR A
F31:            CALL CLS_LCD
                LD HL,MESSAGE+1C0H
                XOR A
                CALL DISPLAY
                LD A,40H
                CALL GOTO
                CALL INPUT_KEYDATA
                CP 00
                RET NZ
                CALL CHK_INPUT_CODE
                CP 00
                JR NZ, F31
                CALL CLS_LCD
                LD HL,MESSAGE+1D0H
                CALL SET_DIS_CODE
                LD IX,CARD_DATA_EXT
                LD HL,TEMP+1F2H
                LD (HL),0FFH
                CALL CHK_IDCODE
                CP ESC
                JR Z, F31
                CP 00
                JR NZ,03
                LD (TEMP+1F2H),A
F32:            CALL T_ADDRESS
                LD HL,MESSAGE+1E0H
                CALL SET_DIS_CODE
                CALL SET_DIS_TER
                CALL CHK_IDCODE
                CP ESC
                JR Z, F31
                CP 00
                JR NZ,03
                LD (TEMP+1F2H),A
                INC B
                LD A,B
                CP 10H
                JR C, F32
                LD A,(TEMP+1F2H)
                CP 00
                JR Z, F31
                LD HL,MESSAGE+1F0H
                XOR A
                CALL DISPLAY
                CALL WAIT_KEY
                JR F31

CHK_IDCODE:     PUSH BC
CID1:           LD A,(IX+0)
                CP 0FFH
                JR Z, CIDE
                CP 0EFH
                JR Z, CIDE
                LD A,(KEY_BUFFER+10)
                CP (IX+0EH)
                JR NZ, CID2
                LD A,(KEY_BUFFER+11)
                CP (IX+0FH)
                JR Z, CID3
CID2:           LD DE,10H
                ADD IX,DE
                JR CID1
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
CID3:           LD HL,TEMP+1D0H
                XOR A
                CALL DISPLAY
                LD HL,MESSAGE+200H
                LD A,40H
                CALL DISPLAY
CID5:           CALL INPUT_KEY
                LD A,E
                CP 0FFH
                JR NZ,03
                POP BC
                POP HL
                RET
                CP ESC
                JR Z, CIDD
                CP YES
                JR Z, CID4
                CP NO
                JR NZ, CID5
                XOR A
                JR CIDD
CID4:           CALL DEL_CARD
                XOR A
CIDD:           POP BC
                RET
CIDE:           LD A,0FFH
                JR CIDD

DEL_CARD:       PUSH BC
                PUSH IX
                PUSH IX
                POP HL
                POP DE
                ADD HL,BC
DC1:            LD A,(HL)
                CP 0FFH
                JR Z, DCD
                CP 0EFH
                JR Z, DCD
                LD BC,10H
                LDIR
                JR DC1
DCD:            LD B,10H
DCDL:           LD A,0FFH
                INC DE
                DJNZ DCDL
                POP BC
                RET

SET_DIS_CODE:   PUSH BC
                LD DE,TEMP+1D0H
                LD BC,10H
                LDIR
                LD HL,KEY_BUFFER
                LD DE,TEMP+1D1H
                LD BC,4
                LDIR
                POP BC
                RET

SET_DIS_TER:    LD A,B
                INC A
                DAA
                PUSH AF
                LD HL,TEMP+1DEH
                SRL A
                SRL A
                SRL A
                SRL A
                ADD A,30H
                LD (HL),A
                INC HL
                ADD A,30H
                LD (HL),A
                RET

T_ADDRESS:      LD DE,TERMINAL_CFG
                LD E,B
                SLA E
                SLA E
                SLA E
                PUSH DE
                POP IX
                LD D,(IX+10)
                LD E,(IX+11)
                PUSH DE
                POP IX
                RET

;       ----- FUNCTION 4 (Set Time) -----

FUNCTION4:      XOR A
F41:            CALL CLS_LCD
                LD HL,MESSAGE+210H
                XOR A
                CALL DISPLAY
                CALL DIS_OLD_WEEK
                CALL INPUT_KEYDATA
                RET NZ
                CALL CHK_INPUT_WEEK
                JR NZ, F41
                LD IY,MESSAGE+220H
                LD IX,DAY_COMPARE
                LD DE,SET_TIME_BUF+1
                LD C,YEAR10
F42:            CALL CLS_LCD
                PUSH IY
                POP HL
                XOR A
                PUSH DE
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                CALL DISPLAY
                CALL DIS_OLD_YMD
                CALL INPUT_KEYDATA
                POP DE
                OR A
                RET NZ
                CALL CHK_INPUT_YMD
                OR A
                JR NZ, F42
                PUSH BC
                INC IX
                INC IX
                LD BC,10H
                ADD IY,BC
                POP BC
                DEC C
                LD A,C
                CP MIN1
                JR NC, F42
                CALL SET_TIME
                RET

CHK_INPUT_YMD:  PUSH IY
                LD IY,KEY_BUFFER
                LD A,(IY+0)
                CP 20H
                JR NZ, CY2
                LD A,(TEMP+1F4H)
                LD (DE),A
                INC DE
                LD A,(TEMP+1F5H)
                LD (DE),A
CY1:            INC DE
                XOR A
CYD:            POP BC
                POP IY
                RET
CY2:            LD A,(IY+1)
                CP 20H
                JR NZ, CY3
                LD A,(IY+0)
                LD (IY+1),A
                XOR A
                LD (IY+0),A
                JR CY4
CY3;            LD A,(IY+2)
                CP 20H
                JR NZ, CYE
CY4:            CALL SHIFT_DATA
                CP (IX+0)
                JR C, CYE
                CP (IX+1)
                JR NC, CYE
                LD A,(IY+0)
                LD (DE),A
                INC DE
                LD A,(IY+1)
                LD (DE),A
                JR CY1
CYE:            LD A,0FFH
                JR CYD

SHIFT_DATA:     LD B,(IY+0)
                SLA B
                SLA B
                SLA B
                SLA B
                LD A,(IY+1)
                AND A,0FH
                OR A,B
                RET

CHK_INPUT_WEEK: LD IY,KEY_BUFFER
                LD A,(IY+0)
                CP 20H
                JR NZ, CW2
                LD A,(TEMP+1F4H)
CWD:            LD (SET_TIME_BUF),A
                RET
CW2:            LD A,(IY+1)
                CP 20H
                JR NZ, CWE
                LD A,(IY+0)
                CP 37H
                JR NC, CWE
                JR CWD
CWE:            LD A,0FFH
                RET

DIS_OLD_WEEK:   PUSH DE
                CALL DIS11
                IN A,(WEEK)
                LD E,A
                LD (TEMP+1F4H),A
                CALL WR_BYTE
                CALL DIS12
                RET

DIS11:          LD A,40H
                CALL GOTO
                LD E,28H
                CALL WR_BYTE
                RET
DIS12:          LD E,29H
                CALL WR_BYTE
                LD E,3AH
                CALL WR_BYTE
                RET

DIS_OLD_YMD:    PUSH DE
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                CALL DIS11
                IN A,(C)
                AND A,3FH
                LD E,A
                LD (TEMP+1F4H),A
                CALL WR_BYTE
                IN A,(C)
                AND A,3FH
                LD E,A
                LD (TEMP+1F5H),A
                CALL WR_BYTE
                INC C
                CALL DIS12
                POP DE
                RET

;  -- -- Computer Program (send data to computer) ------

DAY_BUFFER      EQU 8420H
BYTE_RETRY_SEND EQU 8408H
BYTE_TIMER      EQU 8409H
BYTE_SECOND2    EQU 840BH
ETB             EQU 17H
ENQ             EQU 05H
COMPUTER:       LD HL,MESSAGE+280H
                LD A,40H
                CALL DISPLAY
                IN A,(PORT_OUT)
                SET 1,A
                OUT (PORT_OUT),A
                LD A,03
                CALL SET_TIMER
COM1:           CALL TIMER_CHK
                OR A
                JR NZ, COM1
                CALL RESET_ERROR
                LD A,120
                CALL SET_TIMER
COM2:           INO A,(STAT0)
                BIT 7,A
                JR NZ, COM3
                CALL TIMER_CHK
                JR NZ, COM2
COME:           LD HL,BYTE_RETRY_SEND
                DEC (HL)
                SET 7,(HL)
COMD:           IN A,(PORT_OUT)
                RES 1,A
                OUT (PORT_OUT),A
                LD HL,MESSAGE+2F0H
                LD A,40H
                CALL DISPLAY
                RET
COM3:           AND A,70H
                JR NZ, COM4
                INO A,(RDR0)
                CP ENQ
                JR Z, COM5
                JR 03
COM4:           CALL RESET_ERROR
                LD A,ENQ
                OUTO (TDR0),A
                JR COM2
COM5:           CALL SEND_DATA_COM
                JR NZ, COME
                LD (BYTE_RETRY_SEND),A
                CALL DEL_WORK_TIME
                JR COMD

RESET_ERROR:    LD A,66H
                OUTO (CNTLA0),A
                INO A,(RDR0)
                RET

SEND_DATA_COM:  CALL RESET_ERROR
                CALL INPUT_DAY
                LD HL,STATUS_BYTE
                SET 7,(HL)
                XOR A
                LD (TEMP+1F6H),A
                CALL T_ADDRESS
SC1:            LD HL,DATA_SEND_BUF
                LD C,16
SC3:            LD A,(IX+0)
                CP 0FFH
                JR Z, SC2
                CP 0EFH
                JR Z, SC2
                CALL SC_P1
                LD A,C
                OR A
                JR NZ, SC3
SC9:            LD (HL),ETX
                CALL CRC_WRITE
                LD (HL),ETB
SC4:            CALL RS232_SEND
SC5:            INO A,(STAT0)
                BIT 7,A
                JR NZ, SC6
                CALL TIMER_CHK
                JR NZ,SC5
                LD A,0FFH
                RET
SC6:            INO A,(RDR0)
                CP 11H
                JR NZ, SC7
                LD A,(STATUS_BYTE)
                BIT 7,A
                JR NZ, SC1
                RET
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
SC7:    CP 12H
        JR Z, SC4
        LD A,0FFH
        RET
SC2:    LD A,(TEMP+1F6H)
        INC A
        LD (TEMP+1F6H),A
        CP 10H
        JR C, SC8
        LD A,(STATUS_BYTE)
        RES 7,A
        LD (STATUS_BYTE),A
        LD (HL),ETX
        CALL CRC_WRITE
        LD (HL),EOT
        JR SC4
SC8:    LD B,A
        CALL T_ADDRESS
        JR SC3

SC_P1:  PUSH DE
        LD A,(IX+10)
        CP 0FFH
        JR NZ, SP1
        LD A,(IX+12)
        CP 0FFH
        JR NZ, SP1
        JR SPD
SP1:    LD A,(IX+9)
        CALL SC_P2
        LD DE,DAY_BUFFER+10
        EX DE,HL
        LD BC,4
        LDIR
        EX DE,HL
        POP BC
        LD A,(IX+10)
        CP 0FFH
        JR NZ,01
        XOR A
        CALL SC_P2
        LD A,(IX+11)
        CALL SC_P2
        LD A,(IX+12)
        CP 0FFH
        JR NZ,01
        XOR A
        CALL SC_P2
        LD A,(IX+13)
        CALL SC_P2
        DEC C
SPD:    LD DE,10H
        RET
SC_P2:  PUSH AF
        SRL A
        SRL A
        SRL A
        SRL A
        ADD A,30H
        LD (HL),A
        RET

;       ----- PROGRAM LCD -----

;       ** Init LCD **
P_COMND     EQU C0H
P_WRDATA    EQU C2H
P_RDBUSY    EQU C4H
P_RDDATA    EQU C6H
INIT_LCD:   LD A,38H
            OUT (P_COMND),A
            CALL DELAYL
            CALL DELAYL
            LD A,0FH
            OUT (P_COMND),A
            CALL DELAYL
            LD A,06
            OUT (P_COMND),A
            CALL DELAYL
CLS_LCD:    LD A,01
            OUT (P_COMND),A
            CALL READ
            RET
DELAYL:     PUSH BC
            PUSH AF
            LD B,00H
DELAYL1:    DJNZ DELAYL1
            POP AF
            POP BC
            RET

;       ** DISPLAY **

DISPLAY:    PUSH BC
            CALL GOTO
            LD B,10H
WRL:        LD E,(HL)
            CALL WR_BYTE
            INC HL
            DJNZ WRL
            POP BC
            RET
GOTO:       SET 7,A
            OUT (P_COMND),A
            CALL READ
            RET
WR_BYTE     LD A,E
            OUT (P_WRDATA),A
            CALL READ
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                RET
READ:           IN A,(P_RDBUSY)
                BIT 7,A
                JR NZ,READ
                RET

DEL_BYTE:       LD A,10H
                OUT (P_COMND),A
                CALL READ
                LD E,20H
                CALL WR_BYTE
                LD A,10H
                OUT (P_COMND),A
                CALL READ
                RET

READ_DATA:      PUSH HL
                LD HL,KEY_BUFFER
                CALL GOTO
R1:             CALL RD_BYTE
                LD (HL),A
                CP 20H
                JR Z,03
                INC HL
                JR R1
                POP HL
                RET
RD_BYTE:        IN A,(P_RDDATA)
                PUSH AF
                CALL READ
                POP AF
                RET

RD_ADDRESS:     IN A,(P_RDBUSY)
                RES 7,A
                PUSH AF
                CALL READ
                POP AF
                RET
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
; -------------- Message to Display in LCD --------------

                ORG 1B00H
MESSAGE         DB " INPUT ERROR !  "          ; +00
                DB "Select Function!"          ; +10H
                DB "TIME            "          ; +20H
                DB "Edit CENTER ?   "          ; +30H
                DB "Edit TERMINAL ? "          ; +40H
                DB "TIME:OPEN Center"          ; +50H
                DB "TIME:OFF Center "          ; +60H
                DB "TIME:Send Data  "          ; +70H
                DB "Sel.Term.(1-16) "          ; +80H
                DB "Enter Mode (1-3)"          ; +90H
                DB "TIME:Begin WK-IN"          ; +A0H
                DB "TIME:End WK-IN  "          ; +B0H
                DB "TIME:Begin W-OUT"          ; +C0H
                DB "TIME:End WK-OUT "          ; +D0H
                DB "Insert new card "          ; +E0H
                DB "Card error !    "          ; +F0H
                DB "Press any key ! "          ; +100H
                DB "Select type card"          ; +110H
                DB "TYPE:Check time?"          ; +120H
                DB "TYPE:Extra card?"          ; +130H
                DB "New Card !      "          ; +140H
                DB "Old Card !      "          ; +150H
                DB "Card Full !     "          ; +160H
                DB "PASS CODE ?     "          ; +170H
                DB "PERSON CODE ?   "          ; +180H
                DB "Terminal(  )    "          ; +190H
                DB "Can Pass? (Y,N) "          ; +1A0H
                DB "Remove old Card "          ; +1B0H
                DB "ID CODE (to del)"          ; +1C0H
                DB "(    )Extra Card"          ; +1D0H
                DB "(    )terminl:  "          ; +1E0H
                DB "CODE not found! "          ; +1F0H
                DB "Delete? (Y,N)   "          ; +200H
                DB "Day of week(0-6)"          ; +210H
                DB "Year (00-99)    "          ; +220H
                DB "Month (1-12)    "          ; +230H
                DB "Day (1-31)      "          ; +240H
                DB "Hour (00-23)    "          ; +250H
                DB "Minute (00-59)  "          ; +260H
                DB "Computer Error! "          ; +270H
                DB "Send Data Com.! "          ; +280H

                ORG 1E00H
MESSAGEF        DB "(F1) Edit Config"
                DB "(F2) Add Card   "
                DB "(F3) Delete Card"
                DB "(F4) Set Time   "
                DB "(F5) No Function"
                DB "(F6) No Function"

                ORG 1FF0H
KEY_CODE        DB 0DH,4EH,30H,59H,08H,39H,38H,37H
                DB 1BH,36H,35H,34H,46H,33H,32H,31H
                END
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
;------- Program Main Terminal -------
                ORG 0000H
STACK_POINTER   EQU 9FFFH
PORT_CONT       EQU 43H
PORT_IN         EQU 41H
PORT_KEY        EQU 42H
PORT_OUT        EQU 40H
PSRL_CONT       EQU C1H
PSRL_DATA       EQU C0H
STATUS_BYTE     EQU 9500H
START:          XOR A
                LD HL,0
ST1:            DEC HL
                LD A,L
                OR H
                JR NZ, ST1
                LD SP,STACK_POINTER
                LD A,8AH
                OUT (PORT_CONT),A
                LD A,07FH
                OUT (PSRL_CONT),A
                LD A,15H
                OUT (PSRL_CONT),A
                LD A,07
                OUT (PORT_KEY),A
                IN A,(PORT_KEY)
                BIT 5,A
                JP Z,8000H
                BIT 4,A
                JP NZ,0080H
LOAD:           LD HL,8000H
                LD BC,8000H
                XOR A
                OUT (PORT_OUT),A
                LD A,15H
                OUT (PSRL_CONT),A
INT1:           IN A,(PSRL_CONT)
                BIT 1,A
                JR Z,INT1
                IN A,(PSRL_DATA)
                LD (HL),A
                INC HL
INT2:           IN A,(PSRL_CONT)
                BIT 1,A
                JR NZ,INT3
                DEC BC
                LD A,B
                OR A,C
                JR NZ,INT2
                HALT
INT3:           IN A,(PSRL_DATA)
                LD (HL),A
                INC HL
                LD BC,8000H
                JR INT2

                ORG 0080H
                CALL INIT_LCD
                IN A,(PORT_IN)
                BIT 0,A
                JR Z,MPS
                LD A,19H
                OUT (PORT_OUT),A
MP0:            IN A,(PORT_IN)
                BIT 0,A
                JR Z,MPS
                JR MP0
MPS:            XOR A
                OUT (PORT_OUT),A
                LD A,11H
                LD (STATUS_BYTE),A
                LD A,3
                LD (MISS_CODE_BYTE),A
                LD HL,TIME_BUFFER
                LD B,4
MP:             LD (HL),30H
                INC HL
                DJNZ MP
                CALL PRO_D1
MP1:            JP PRO_8251
M_8251:         LD HL,STATUS_BYTE
                BIT 4,(HL)
                JR NZ, MP1
                BIT 3,(HL)
                JR NZ, MP1
                BIT 5,(HL)
                JR NZ, MP1
                JP PRO_CARD

;       ** PROGRAM_D1 **
DATA_SEND_BUF           EQU 9000H
DATA_RECEIVE_BUF        EQU 9100H
ADDRESS         EQU 80H
STX             EQU 02H
ETX             EQU 03H
EOT             EQU 04H
PRO_D1:         LD HL,DATA_SEND_BUF
                LD (HL),ADDRESS
                INC HL
                LD (HL),0D1H
                CALL CRC_WRITE
                LD (HL),EOT
                RET

;       ---- PROGRAM CARD ----

POWER_ON_TIME           EQU 0100H
DELAY_DIS       EQU 05H
CARD_COUNT_BUF          EQU 9200H
POWER_ON_BYTE           EQU 9503H
MISS_CODE_BYTE          EQU 9502H
MESSAGE         EQU 1E00H
MESSAGE_1       EQU MESSAGE+0
MESSAGE_2       EQU MESSAGE+10H
MESSAGE_3       EQU MESSAGE+20H
MESSAGE_4       EQU MESSAGE+30H
MESSAGE_5       EQU MESSAGE+40H
MESSAGE_6       EQU MESSAGE+50H
MESSAGE_7       EQU MESSAGE+60H
MESSAGE_8       EQU MESSAGE+70H
MESSAGE_9       EQU MESSAGE+80H
MESSAGE_10      EQU MESSAGE+90H
MESSAGE_11      EQU MESSAGE+0A0H
MESSAGE_12      EQU MESSAGE+0B0H
MESSAGE_13      EQU MESSAGE+0C0H
MESSAGE_14      EQU MESSAGE+0D0H
MESSAGE_15      EQU MESSAGE+0E0H
MESSAGE_16      EQU MESSAGE+0F0H

PRO_CARD:       IN A,(PORT_IN)
                BIT 1,A
                JR NZ,PC2
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                LD HL,STATUS_BYTE
                BIT 6,(HL)
                JR Z, PC5
PC2:            CALL DIS_CARD_READ
                CALL CARD_COUNT
                OR A,A
                JR NZ,DIS_CARD_ERR
                CALL PROG_12
                OR A,A
                JR NZ,DIS_CARD_ERR
                CALL CARD_CRC_CHK
                OR A,A
                JR NZ,DIS_CARD_ERR
                CALL BEEPL
                CALL CHECK_IN_OUT
                OR A
                JR NZ,CARD_OUT
                LD HL,STATUS_BYTE
                BIT 1,(HL)
                JR Z, PC3
CHK_CODE:       CALL CHECK_CODE
                OR A
                JR NZ,CARD_OUT
PC3:            CALL PRO_D3
                LD HL,STATUS_BYTE
                SET 5,(HL)
                JR PC5
DIS_CARD_ERR:   CALL CLS_LCD
                LD A,00
                LD HL,MESSAGE_4
                CALL DISPLAY
CARD_OUT:       LD HL,STATUS_BYTE
                RES 5,(HL)
                RES 6,(HL)
                LD A,03
                LD (MISS_CODE_BYTE),A
                CALL PRO_D5
                CALL DELAYD
                CALL DIS_READY
PC5:            JP PRO_8251
DIS_READY:      LD HL,POWER_ON_TIME
                LD (POWER_ON_BYTE),HL
                CALL CLS_LCD
                CALL DISPLAY_TIME
                LD A,00
                LD HL,MESSAGE_6
                CALL DISPLAY
                RET
DIS_CARD_READ:  CALL CLS_LCD
                LD A,00H
                LD HL,MESSAGE_5
                CALL DISPLAY
                RET
DELAYD:         LD B,DELAY_DIS
DELAYD2:        LD HL,0
DELAYD1:        DEC HL
                LD A,L
                OR A,H
                JR NZ,DELAYD1
                DJNZ DELAYD2
                RET

BEEP:           PUSH BC
                LD C,4
BEEP2:          LD B,80H
                IN A,(PORT_OUT)
BEEP1:          SET 6,A
                OUT (PORT_OUT),A
                CALL FBEEP
                RES 6,A
                OUT (PORT_OUT),A
                CALL FBEEP
                DJNZ BEEP1
                DEC C
                JR NZ,BEEP2
                POP BC
                RET
BEEPL:          CALL BEEP
                CALL BEEP
                RET
FBEEP:          PUSH BC
                LD B,20H
FBEEP1          DJNZ FBEEP1
                POP BC
                RET

;          ----- PROGRAM CARD COUNT -----

CARD_COUNT:     LD HL,CARD_COUNT_BUF
                LD C,00H
                LD B,0FFH
CC1:            INC C
                JP Z, CC4
                IN A,(PORT_IN)
                BIT 1,A
                JP NZ, CC2
                LD B,00
                JP CC1
CC2:            RLC B
                JP C, CC1
                LD B,0FFH
                LD (HL),C
                INC HL
                LD C,00
                JP CC1
CC4:            LD A,00
                RET

;          --- PROGRAM DECODE CARD DATA ---

COUNT_BUFH      EQU 92H
CARD_DAT_BUF    EQU 9200H
PROG_12:        PUSH IX
                PUSH IY
                LD IX,CARD_COUNT_BUF+5
                LD IY,CARD_DAT_BUF
                LD E,00H
                LD H,00H
                LD B,00H
                LD L,05H
                LD C,28H
                LD A,(IX+0)
                SRL A
                SRL A
                LD D,A
                SRL A
                ADD A,D
                LD D,A
DC1:            PUSH BC
                PUSH IX
                POP BC
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                LD A,COUNT_BUFH
                CP B
                JR NC, DC11
                LD A,7FH
                CP C
                JR NC, DC11
                POP BC
                JP DC_ERR
DC11:           LD A,(IX+0)
                LD B,(IX+1)
                SUB A,B
                JR NC,02H
                NEG
                POP BC
                CP D
                JR NC, DC2
                SRL E
                JR C, DC3
DC4:            LD A,(IX+1)
                SRL A
                SRL A
                LD D,A
                SRL A
                ADD A,D
                LD D,A
                INC IX
                RRC H
                JR NC, DC1
                SRL B
                LD E,00H
DC5:            DEC L
                JR NZ, DC1
                SRL B
                SRL B
                SRL B
                PUSH BC
                LD C,04H
                LD A,00H
                JR DC7
DC2:            SRL E
                JR NC, DC6
                PUSH BC
                LD A,(IX+0)
                LD B,(IX+1)
                CP B
                POP BC
                JR C, DC4
                JR DC_ERR
DC6:            PUSH BC
                LD A,(IX+0)
                LD B,(IX+1)
                CP B
                POP BC
                JR C, DC_ERR
DC3:            PUSH BC
                LD A,(IX+1)
                LD B,(IX+2)
                SUB A,B
                JR NC,02H
                NEG
                POP BC
                CP D
                JR NC, DC_ERR
                LD D,(IX+2)
                SRL D
                INC IX
                INC IX
                LD H,0FFH
                RRC H
                RR B
                LD E,11H
                JR DC5
DC7:            SRL B
                JR NC, DC8
                INC A
DC8:            DEC C
                JR NZ, DC7
                AND 01H
                CP B
                POP BC
                JR Z, DC_ERR
                RES 4,B
                LD (IY+0),B
                INC IY
                DEC C
                JR Z, DC_OK
                LD A,0FH
                CP B
                JR NZ,02H
                LD C,01H
                LD L,05H
                JP DC1
DC_OK:          LD A,00H
                JR DC_END
DC_ERR:         LD A,0FFH
                LD (IY+0),0EEH
                INC IY
                DEC C
                JR NZ, DC_ERR
DC_END:         POP IY
                POP IX
                RET

;               ----- CARD_CRC_CHECK -----

CARD_CRC_CHK:   LD HL,CARD_DAT_BUF
                LD B,00H
                LD C,28H
                LD A,0BH
                CP (HL)
                JR NZ, CK_ERR
CK1:    DEC C
                JR Z, CK_ERR
                LD A,B
                XOR (HL)
                LD B,A
                LD A,0FH
                CP (HL)
                JR Z, CK2
                INC HL
                JR CK1
CK2:            INC HL
                LD A,B
                CP (HL)
                JR NZ, CK_ERR
                LD A,00H
                RET
CK_ERR: LD A,0FFH
                RET
;               ----- CHECK_IN_OUT -----
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
SCANKEY_ADDH          EQU 1FH
BYTE_RELS_KEY         EQU 9505H
BYTE_WAITK      EQU 9506H
BYTE_WAITKL     EQU 9508H
IN              EQU 59H
OUT             EQU 4EH
YES             EQU 59H
NO              EQU 4EH
CANCEL          EQU 'C'
IN_OUT_BUF      EQU 9228H

CHECK_IN_OUT:   CALL CLS_LCD
                LD A,00H
                LD HL,MESSAGE_8
                CALL DISPLAY
IO1:            CALL INPUT_KEY
                LD A,E
                CP IN
                JR Z, IO2
                CP OUT
                JR Z, IO2
                CP CANCEL
                JR Z, IO3
                CP 0FFH
                JR Z, IO3
                JR IO1
IO2:            LD (IN_OUT_BUF),A
                XOR A
                RET
IO3:            CALL CLS_LCD
                LD A,00
                LD HL,MESSAGE_16
                CALL DISPLAY
                LD A,0FFH
                RET

;          -------- CHECK_CODE --------

CODE_BUF        EQU 9229H
CHECK_CODE:     XOR A
CO1:            LD IX,CODE_BUF
                CALL DIS_ENTER_CODE
                LD B,04
CO2:            CALL INPUT_KEY
                LD A,E
                CP 30H
                JR C, CO3
                CP 40H
                JR NC, CO3
                LD (IX+0),A
                CALL DIS_X
                INC IX
                DJNZ CO2
                LD A,00H
                RET
CO3:            CP CANCEL
                JR Z, COE
                CP 0FFH
                JR Z, COE
                JR CO2
COE:            CALL CLS_LCD
                LD A,00
                LD HL,MESSAGE_16
                CALL DISPLAY
                LD A,01FFH
                RET

DIS_ENTER_CODE: CALL CLS_LCD
                PUSH HL
                LD A,00H
                LD HL,MESSAGE_1
                CALL DISPLAY
                LD A,40H
                CALL GOTO
                POP HL
                RET
DIS_X:          PUSH BC
                LD E,78H
                CALL WR_BYTE
                POP BC
                RET
DIS_YN:         PUSH HL
                LD A,00H
                LD HL,MESSAGE_2
                CALL DISPLAY
                LD A,40H
                LD HL,MESSAGE_3
                CALL DISPLAY
                POP HL
                RET

;          ------ PROGRAM SCANKEY ------

WAITK_TIME      EQU 500H
INPUT_KEY:      PUSH HL
                PUSH BC
                LD HL,WAITK_TIME
                LD (BYTE_WAITK),HL
                XOR A
                LD (BYTE_WAITKL),A
IK1:            CALL SCANKEY
                OR A,A
                JR Z, IK2
                LD HL,BYTE_WAITKL
                DEC (HL)
                JR NZ, IK1
                LD HL,(BYTE_WAITK)
                DEC HL
                LD (BYTE_WAITK),HL
                LD A,L
                OR A,H
                JR NZ, IK1
                LD E,0FFH
                JR IKD
IK2:            LD E,(HL)
                CALL BEEP
IKD:            POP BC
                POP HL
                RET

SCANKEY:        PUSH BC
                PUSH DE
                LD B,4
                LD D,0EEH
                LD C,PORT_KEY
SK1:            OUT (C),D
                IN E,(C)
                LD A,E
                AND A,0F0H
                LD E,A
                CP 0F0H
                JR NZ, SK3
                RLC D
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                DJNZ SK1
                LD A,(BYTE_RELS_KEY)
                OR A,A
                JR Z, SKD
                CALL DELAYK
                XOR A
                LD (BYTE_RELS_KEY),A
                JR SKD
SK3:            LD A,(BYTE_RELS_KEY)
                OR A,A
                JR NZ, SKD
                XOR A
SK4:            SRL D
                JR NC,03
                INC A
                JR SK4
                SLA A
                SLA A
SK5:            SLA E
                JR NC,03
                INC A
                JR SK5
                OR A,0F0H
                LD L,A
                LD H,SCANKEY_ADDH
                LD A,0FFH
                LD (BYTE_RELS_KEY),A
                XOR A
                JR 02
SKD:            LD A,0FFH
                POP DE
                POP BC
                RET

DELAY_TIME      EQU 5
DELAYK:         PUSH BC
                LD C,DELAY_TIME
                LD B,00H
DK1:            NOP
                NOP
                DJNZ DK1
                DEC C
                JR NZ,DK1
                POP BC
                RET

        ** PROGRAM_D3 **
PRO_D3:         LD DE,CARD_DAT_BUF+1
                LD HL,DATA_SEND_BUF
                LD B,10H
                LD (HL),ADDRESS
                INC HL
                LD (HL),0D3H
                INC HL
                LD (HL),STX
H1:             INC HL
                LD A,(DE)
                CP 0FH
                JR Z,H2
                ADD A,30H
                LD (HL),A
                INC DE
                INC DE
                DJNZ H1
                INC HL
H2:             LD (HL),1DH
                INC HL
                LD DE,IN_OUT_BUF
                LD A,(DE)
                LD (HL),A
                INC HL
                INC DE
                LD A,(STATUS_BYTE)
                BIT 1,A
                JR Z,H4
                LD B,04H
H3:             LD A,(DE)
                LD (HL),A
                INC DE
                INC HL
                DJNZ H3
H4:             LD (HL),ETX
                CALL CRC_WRITE
                LD (HL),EOT
                RET

;       -------- PROGRAM_D5 --------

PRO_D5:         LD HL,DATA_SEND_BUF
                LD (HL),ADDRESS
                INC HL
                LD (HL),0D5H
                CALL CRC_WRITE
                LD (HL),EOT
                LD HL,CARD_COUNT_BUF
                LD BC,0230H
L1:             LD (HL),00H
                INC HL
                DEC BC
                LD A,B
                OR A,C
                JR NZ,L1
                RET

;       -------- Program serial data --------

SYNC            EQU 16H
ALL_ADDRESS     EQU 0A0H
PRO_8251:       IN A,(PSRL_CONT)
                BIT 1,A
                JR Z,A3
                AND A,28H
                JR NZ,A3
                IN A,(PSRL_DATA)
                CP SYNC
                JR NZ,A3
                LD B,80H
A1:             CALL PULSE_DETECT
                IN A,(PSRL_CONT)
                BIT 1,A
                JR NZ,A6
                DEC B
                JR Z,A3
                JR A1
A6:             AND A,28H
                JR NZ,A3
                IN A,(PSRL_DATA)
                CP ALL_ADDRESS
                JR Z,A4
                CP ADDRESS
                JR NZ,A3A4:
                LD HL,DATA_RECEIVE_BUF
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                LD (HL),A
                INC L
A7:             LD B,80H
A2:             CALL PULSE_DETECT
                IN A,(PSRL_CONT)
                BIT 1,A
                JR NZ,A8
                DEC B
                JR Z,A3
                JR A2
A8:             AND A,28H
                JR NZ,A3
                IN A,(PSRL_DATA)
                LD (HL),A
                CP EOT
                JR Z,A9
                INC L
                JR Z,A3
                JR A7
A9:             CALL CRC_CHECK
                CP 00H
                JR NZ,A3
                LD HL,STATUS_BYTE
                RES 3,(HL)
                LD A,(DATA_RECEIVE_BUF+1)
                CP 0D0H
                JP Z, PRO_D0
                CP 0D2H
                JP Z, PRO_D2
                CP 0D4H
                JP Z, PRO_D4
                CP 0D6H
                JP Z, PRO_D6
                CP 0D8H
                JP Z, PRO_D8
A3:             IN A,(PORT_OUT)
                RES 5,A
                OUT (PORT_OUT),A
                LD A,15H
                OUT (PSRL_CONT),A
                JP M_8251
A_8251:         JR A3
PULSE_DETECT:   PUSH HL
                LD HL,STATUS_BYTE
                BIT 5,(HL)
                JR NZ,M1
                BIT 6,(HL)
                JR NZ,M1
                IN A,(PORT_IN)
                BIT 1,A
                JR Z,M1
                SET 6,(HL)
M1:             POP HL
                RET

;               -------- Program Data Send ---  ---

DATA_SEND:      LD HL,DATA_SEND_BUF
                LD A,15H
                OUT (PSRL_CONT),A
                IN A,(PORT_OUT)
                SET 5,A
                OUT (PORT_OUT),A
                LD A,16H
                OUT (PSRL_DATA),A
B1:             IN A,(PSRL_CONT)
                BIT 0,A
                JR Z,B1
                LD A,(HL)
                OUT (PSRL_DATA),A
                CP 04H
                JR Z,B2
                INC HL
                JR B1
B2:             IN A,(PSRL_CONT)
                BIT 2,A
                JR Z,B2
                IN A,(PORT_OUT)
                RES 5,A
                OUT (PORT_OUT),A
                RET

;               ------ TIME_PROGRAM (D0) ------

TIME_BUFFER     EQU 9510H
CLOSE_DOOR_     BYTE EQU 9501H
PRO_D0:         LD DE,TIME_BUFFER
                LD HL,DATA_RECEIVE_BUF+3
A01:            LD A,(HL)
                CP 0EH
                JR Z,A02
                CP EOT
                JR Z,A06
                INC HL
                JR A01
A02:            LD B,03H
A03:            INC HL
                LD A,(HL)
                LD (DE),A
                INC DE
                DJNZ A03
                INC HL
                LD A,(DE)
                CP (HL)
                JR Z,A04
                LD A,(HL)
                LD (DE),A
                LD HL,STATUS_BYTE
                BIT 5,(HL)
                JR NZ,A04
                CALL DISPLAY_TIME
A04:            LD HL,CLOSE_DOOR_BYTE
                BIT 4,(HL)
                JR Z,POWER_OFF
                CALL CLOSE_DOOR
                JR A06
POWER_OFF:      LD HL,(POWER_ON_BYTE)
                DEC HL
                LD (POWER_ON_BYTE),HL
                LD A,L
                OR H
                JR NZ,A06
                CALL BEEPL
                LD A,80H
                OUT (PORT_OUT),A
                HALT
A06:            JP A_8251
CLOSE_DOOR:     DEC (HL)
                LD A,(HL)
                AND A,0FH
                RET NZ
                IN A,(PORT_IN)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                BIT 0,A
                JR Z, CD1
                IN A,(PORT_OUT)
                OR A,19H
                OUT (PORT_OUT),A
CD2:            IN A,(PORT_IN)
                BIT 0.A
                JR Z, CD1
                JR CD2
CD1:            IN A,(PORT_OUT)
                RES 4,A
                RES 0,A
                OUT (PORT_OUT),A
                RES 4,(HL)
                LD HL,STATUS_BYTE
                BIT 2,(HL)
                RET Z
                SET 1,(HL)
                RET

;       ------ DISPLAY TIME PROGRAM ------

DISPLAY_TIME:   LD HL,MESSAGE_13
                LD A,40H
                CALL GOTO
                LD C,05H
B01:            LD E,(HL)
                CALL WR_BYTE
                INC HL
                DEC C
                JR NZ,B01
                LD HL,TIME_BUFFER
                LD C,05H
B02:            LD E,(HL)
B03:            CALL WR_BYTE
                LD A,C
                CP 04H
                JR Z,B04
                INC HL
                DEC C
                JR NZ,B02
                RET
B04:            LD E,3AH
                DEC C
                JR B03

;
;       ------ PROGRAM DATA SEND (D2) ------

PRO_D2:         CALL DATA_SEND
                JP A_8251

;
;---* PROGRAM WAIT DATA FROM CENTER (D4) ---

PRO_D4:         CALL DATA_SEND
                LD HL,STATUS_BYTE
                SET 3,(HL)
                JP A3

;
;       -------- PROGRAM SET MODE (D6) --------

MODE1           EQU 31H
MODE2           EQU 32H
MODE3           EQU 33H
PRO_D6:         LD HL,DATA_RECEIVE_BUF+3
A61:            LD A,(HL)
                CP 0FH
                JR Z,A62
                CP EOT
                JR Z,A67
                INC HL
                JR A61
A62:            INC HL
                LD A,(HL)
                LD HL,STATUS_BYTE
                CP MODE1
                JR NZ,A63
                RES 1,(HL)
                JR A65
A63:            CP MODE2
                JR NZ,A64
                SET 1,(HL)
                JR A65
A64:            CP MODE3
                JR NZ,A66
                SET 1,(HL)
                SET 2,(HL)
                JR A66
A65:            RES 2,(HL)
A66:            RES 4,(HL)
A67:            CALL PRO_D5
                CALL DIS_READY
                JP A_8251

;
;       ---------- PROGRAM (D8)----------
*
DIS_DELAY       EQU 04H
DOOR_OPEN_TIME          EQU 18H
PRO_D8:         LD HL,DATA_RECEIVE_BUF+3
A81:            LD A,(HL)
                CP 10H
                JR Z,A82
                CP EOT
                JP Z,A_8251
                INC HL
                JR A81
A82:            INC HL
                LD A,(HL)
                LD HL,MISS_CODE_BYTE
                CP 30H
                JR Z,A83
                CP 31H
                JR Z,A84
                CP 32H
                JR Z,A85
                CP 33H
                JR Z,A86
                CP 34H
                JR Z,A87
                CP 35H
                JR Z,A88
                JP A_8251
A83:            LD A,DOOR_OPEN_TIME
                LD (CLOSE_DOOR_BYTE),A
                IN A,(PORT_OUT)
                AND A,0F0H
                OR A,11H
                OUT (PORT_OUT),A
                XOR A
                LD HL,MESSAGE_9
                CALL DISPLAY
                LD HL,STATUS_BYTE
                BIT 2,(HL)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                JR Z,02
                RES 1,(HL)
                JP CARD_OUT
A84:            XOR A
                LD HL,MESSAGE_10
                CALL DISPLAY
                JP CARD_OUT
A85:            LD HL,MISS_CODE_BYTE
                DEC (HL)
                JR Z,A84
                XOR A
                LD HL,MESSAGE_11
                CALL DISPLAY
                LD C,DIS_DELAY
                CALL DELAY8
                JP CHK_CODE
A86:            LD A,40H
                LD HL,MESSAGE_12
                CALL DISPLAY
                JR A83
A87:            XOR A
                LD HL,MESSAGE_14
                CALL DISPLAY
                JP CARD_OUT
A88:            XOR A
                LD HL,MESSAGE_15
                CALL DISPLAY
                JP CARD_OUT
DELAY8:         LD B,00H
DD:             PUSH BC
                LD B,00H
DE1:            NOP
                NOP
                DJNZ DE1
                POP BC
                DJNZ DD
                DEC C
                JR NZ, DELAY8
                RET

;       ----- PROGRAM DATA_SERIAL_CRC -----

CRC_WRITE:      CALL CRC_CAL
                INC HL
                LD (HL),D
                INC HL
                LD (HL),E
                INC HL
                RET

;       ------ PROGRAM CRC_CHECK ------

CRC_CHECK:      DEC HL
                DEC HL
                DEC HL
                CALL CRC_CAL
                INC HL
                LD A,(HL)
                CP D
                JR NZ, CRK1
                INC HL
                LD A,(HL)
                CP E
                JR NZ, CRK1
                LD A,00H
                RET
CRK1:           LD A,0FFH
                RET

;       ------ PROGRAM CRC CALCULATE -----

CRC_CAL:        PUSH HL
                LD B,L
                INC B
                XOR A
                LD D,A
CRL1:           ADD A,(HL)
                JR NC, CRL4
                INC D
CRL4:           DEC HL
                DJNZ CRL1
                LD E,A
                LD A,1FH
                CP D
                JR C,CRL2
                SET 6,D
CRL2:           CP E
                JR C,CRL3
                SET 6,E
CRL3:           POP HL
                RET

;       ---------- Init LCD ----------

P_DATA          EQU 80H
P_SIGN          EQU 82H
P_READ          EQU 84H
INIT_LCD:       LD A,38H
                OUT (P_DATA),A
                CALL DELAYL
                CALL DELAYL
                LD A,0CH
                OUT (P_DATA),A
                CALL DELAYL
                LD A,06
                OUT (P_DATA),A
                CALL DELAYL
CLS_LCD:        LD A,01
                OUT (P_DATA),A
                CALL READ
                RET
DELAYL:         PUSH BC
                PUSH AF
                LD B,00H
DELAYL1:        DJNZ DELAYL1
                POP AF
                POP BC
                RET

;               ---------- DISPLAY ----------
DISPLAY:        PUSH BC
                CALL GOTO
                LD B,10H
WRL:            LD E,(HL)
                CALL WR_BYTE
                INC HL
                DJNZ WRL
                POP BC
                RET
GOTO:           SET 7,A
                OUT (P_DATA),A
                CALL READ
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                RET
WR_BYTE         LD A,E
                OUT (P_SIGN),A
                CALL READ
                RET
READ:           IN A,(P_READ)
                BIT 7,A
                JR NZ,READ
                RET

                ORG 1E00H
MESSAGE_DATA            DB ' Enter CODE
                DB '(YES) to OK
                DB '(NO) to abort
                DB 'CARD Error!
                DB 'Read CARD
                DB 'READY
                DB '                '
                DB '(IN) or (OUT)?  '
                DB 'PASS OK!        '
                DB 'NO PASS!        '
                DB 'CODE Missing!
                DB 'WORK IN OK!
                DB 'TIME
                DB 'WORK OUT OK!
                DB 'WORK OUT Error! '
                DB 'Cancle CARD!

                ORG 1FF0H
KEY_CODE        DB
43H,4EH,59H,30H,0,39H,38H,37H
                DB
0,36H,35H,34H,0,33H,32H,31H
                END
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
= 0005                  ENQ             EQU 05
= 0002                  STX             EQU 02
= 0017                  ETB             EQU 17H
= 0004                  EOT             EQU 04
= 8000                  BUF1_LENGTH     EQU 32*1024
= 0100                  BUF2_LENGTH     EQU 256

0000                    CODE            SEGMENT
                                        ASSUME CS:CODE,DS:CODE
0100                                    ORG 100H
0100                    START:
0100  B4 00                             MOV AH,0
0102  B0 FB                             MOV AL,0FBH
0104  BA 0001                          MOV DX,1
0107  CD 14                             INT 14H
0109  B4 3C                             MOV AH,3CH
010B  B9 0000                          MOV CX,0
010E  BA 0221 R                        MOV DX,OFFSET FTEMP
0111  CD 21                             INT 21H
0113  73 03                             JNC A01
0115  E9 01BD R                        JMP EXIT
0118  A3 022B R               A01:      MOV FT_HANDLE,AX
011B  BF 022D R                        MOV DI,OFFSET BUFFER1
011E  BE 822D R               A11:      MOV SI,OFFSET BUFFER2
0121  B0 05                             MOV AL,ENQ
0123  E8 01E2 R                        CALL SEND_BYTE
0126  B4 02                A1:       MOV AH,2
0128  BA 0001                          MOV DX,1
012B  CD 14                             INT 14H
012D  8A FC                             MOV BH,AH
012F  80 E4 80                          AND AH,80H
0132  74 03                             JZ  A12
0134  E9 01D9 R                        JMP TIME_OUT
0137  80 E7 0E                A12:      AND BH,0EH
013A  74 03                             JZ A13
013C  E9 01C2 R                        JMP D_ERR
013F  3C 02                A13:      CMP AL,STX
0141  74 03                             JZ A2
0143  EB 7D 90                         JMP D_ERR
0146  B4 02                A2:       MOV AH,2
0148  BA 0001                          MOV DX,1
014B  CD 14                             INT 14H
014D  8A FC                             MOV BH,AH
014F  80 E4 80                          AND AH,80H
0152  74 03                             JZ A21
0154  E9 01D9 R                        JMP TIME_OUT
0157  80 E7 0E                A21:      AND BH,0EH
015A  75 66                             JNZ D_ERR
015C  3C 05                             CMP AL,ENQ
015E  74 BE                             JZ A11
0160  88 04                            MOV (SI+0),AL
0162  46                                INC SI
0163  3C 17                            CMP AL,ETB
0165  74 06                            JZ A3
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
0167  3C 04                         CMP AL,EOT
0169  74 02                         JZ A3
016B  EB D9                         JMP A2
016D  A2 832D R             A3:     MOV (TEMP),AL
0170  E8 01EA R                      CALL CRC_CHK
0173  80 FC 00                      CMP AH,0
0176  75 4A                         JNZ D_ERR
0178  8B CE                         MOV CX,SI
017A  81 E9 822D R                   SUB CX,OFFSET BUFFER2
017E  BE 822D R                      MOV SI,OFFSET BUFFER2
0181  8A 24                 L1:     MOV AH,(SI+0)
0183  88 25                         MOV (DI+0),AH
0185  46                            INC SI
0186  47                            INC DI
0187  E2 F8                         LOOP L1
0189  B0 11                         MOV AL,11H
018B  E8 01E2 R                      CALL SEND_BYTE
018E  8A 26 832D R                   MOV AH,(TEMP)
0192  80 FC 04                      CMP AH,EOT
0195  74 05                         JZ A4
0197  BE 822D R                      MOV SI,OFFSET BUFFER2
019A  EB 8A                         JMP A1
019C  8B CF                 A4:     MOV CX,(DI)
019E  81 E9 022D R                   SUB CX,OFFSET BUFFER1
01A2  51                            PUSH CX
01A3  B4 40                         MOV AH,40H
01A5  8B 1E 022B R                   MOV BX,FT_HANDLE
01A9  BA 022D R                      MOV DX,OFFSET BUFFER1
01AC  CD 21                         INT 21H
01AE  72 29                         JC TIME_OUT
01B0  59                            POP CX
01B1  3B C1                         CMP AX,CX
01B3  75 24                         JNZ TIME_OUT
01B5  B4 3E                         MOV AH,3EH
01B7  8B 1E 022B R                   MOV BX,FT_HANDLE
01BB  CD 21                         INT 21H
01BD  B8 4C00               EXIT:   MOV AX,4C00H
01C0  CD 21                         INT 21H
01C2  B4 02                 D_ERR:  MOV AH,2
01C4  BA 0001                       MOV DX,1
01C7  CD 14                         INT 14H
01C9  80 E4 80                      AND AH,80H
01CC  74 F4                         JZ D_ERR
01CE  B0 12                         MOV AL,12H
01D0  E8 01E2 R                      CALL SEND_BYTE
01D3  BE 822D R                      MOV SI,OFFSET BUFFER2
01D6  E9 0126 R                      JMP A1
01D9  B4 41                 TIME_OUT:   MOV AH,41H
01DB  BA 0221 R                      MOV DX,OFFSET FTEMP
01DE  CD 21                         INT 21H
01E0  EB DB                         JMP EXIT
01E2  B4 01                 SEND_BYTE:  MOV AH,1
01E4  BA 0001                       MOV DX,1
01E7  CD 14                         INT 14H
01E9  C3                            RET
01EA  4E                    CRC_CHK:    DEC SI
01EB  4E                            DEC SI
01EC  4E                            DEC SI
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
01ED  8A 3C                                    MOV BH,(SI+0)
01EF  8A 5C 01                                 MOV BL,(SI+1)
01F2  56                                       PUSH SI
01F3  8B CE                                    MOV CX,SI
01F5  81 E9 822D R                             SUB CX,OFFSET BUFFER2
01F9  BE 822D R                                MOV SI,OFFSET BUFFER2
01FC  BA 0000                                  MOV DX,0
01FF  03 14.                          C1:      ADD DX,(SI+0)
0201  46                                       INC SI
0202  E2 FB                                    LOOP C1
0204  80 FA 20                                 CMP DL,20H
0207  73 03                                    JNC C2
0209  80 CA 40                                 OR DL,40H
020C  80 FE 20                        C2:      CMP DH,20H
020F  73 03                                    JNC C3
0211  80 CE 40                                 OR DH,40H
0214  3B D3                           C3:      CMP DX,BX
0216  75 05                                    JNZ C4
0218  B4 00                                    MOV AH,0
021A  5E                              C5:      POP SI
021B  4E                                       DEC SI
021C  C3                                       RET
021D  B4 FF                           C4:      MOV AH,0FFH
021F  EB F9                                    JMP C5
0221  46 54 45 4D 50 2E 41            FTEMP    DB 'FTEMP.ASC',0
      53 43 00
022B  0000                            FT_HANDLE DW 0
022D  8000[00                         BUFFER1  DB BUF1_LENGTH DUP(0)
822D  0100[00                         BUFFER2  DB BUF2_LENGTH DUP(0)
832D  00                              TEMP     DB 0
832E                                  CODE     ENDS
                                               END START

Segments and Groups:

                N a m e                 Size Align Combine Class

CODE . . . . . .                        832E PARA NONE

Symbols:

                N a m e                 Type Value Attr

A01  . . . . . .                        L NEAR 0118 CODE
A1 . . . . . .                          L NEAR 0126 CODE
A11  . . . . . .                        L NEAR 011E CODE
A12  . . . . . .                        L NEAR 0137 CODE
A13  . . . . . .                        L NEAR 013F CODE
A2 . . . . . .                          L NEAR 0146 CODE
A21  . . . . . .                        L NEAR 0157 CODE
A3 . . . . . .                          L NEAR 016D CODE
A4 . . . . . .                          L NEAR 019C CODE

BUF1_LENGTH  . . . . . .                Number 8000
BUF2_LENGTH  . . . . . .                Number 0100
BUFFER1  . . . . . .                    L BYTE  022D CODE Length = 8000
BUFFER2  . . . . . .                    L BYTE  822D CODE Length = 0100

C1 . . . . . .                          L NEAR 01FF CODE
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
C2 . . . . . .  L NEAR 020C CODE
C3 . . . . . .  L NEAR 0214 CODE
C4 . . . . . .  L NEAR 021D CODE
C5 . . . . . .  L NEAR 021A CODE
CRC_CHK . . . . . .  L NEAR 01EA CODE
D_ERR . . . . . .  L NEAR 01C2 CODE

ENQ . . . . . .  Number 0005
EOT . . . . . .  Number 0004
ETB . . . . . .  Number 0017
EXIT . . . . . .  L NEAR 01BD CODE

FTEMP . . . . . .  L BYTE  0221 CODE
FT_HANDLE . . . . . .  L WORD  022B CODE

L1 . . . . . .  L NEAR 0181 CODE

SEND_BYTE . . . . . .  L NEAR 01E2 CODE
START . . . . . .  L NEAR 0100 CODE
STX . . . . . .  Number 0002

TEMP . . . . . .  L BYTE  832D CODE
TIME_OUT . . . . . .  L NEAR 01D9 CODE


    145 Source  Lines
    145 Total   Lines
     54 Symbols

  49278 Bytes symbol space free

      0 Warning Errors
      0 Severe  Errors
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
£££ Program DBMAIN.C FOR AUTOMATIC SALARY CALCULATING £££

£include<stdio.h>
£include<conio.h>
£include<dos.h>
£include<time.h>
£include<ctype.h>
struct dat
 {
 char name[35];
 char sex[6];
        int  age;
 long salary;
 char address[70];
 char position[20];
 char birthday[20];
 char nofbank[20];
 char nofper[20];
 char nofsocial[20];
 char group[3];
 }data[100];
struct ptime
 {
 char code;
 char day;
 char date;
 char hourin;
 char minutin;
 char hourout;
 char minutout;
 } ti;
struct month
 {
 char mon[9];
 } mon;
struct daystop
 {
 char stdate;
 char stmonth;
 } st;
struct config
 {
 char chi;
 char cmi;
 char cho;
 char cmo;
 char cothi;
 char cotmi;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
 char cotho;
 char cotmo;
 int  cotpn;
 int  cotpsat;
 int  cotpsun;
 int  cotpsto;
 char cml;
 int  cmlb;
 } co[];
main()
{
 char ch1;
 initdata();
 loop:
 do
     {
     clrscr();
     printf("\n\n\n\n");
     printf("\t\t      ** โปรแกรม เครื่องคิดเงินเดือนยัดไนมัติ **");
     printf("\n\n");
     printf("\t\t                    ดูรายชื่อพนักงาน : 1\n");
     printf("\t\t                แก้ไขข้อมูลของพนักงาน : 2\n");
     printf("\t\t                          ลบข้อมูล : 3\n");
     printf("\t\t                    ดูเวลาการทำงาน : 4\n");
     printf("\t\t                กำหนดเวลาการทำงาน : 5\n");
     printf("\t\t                    ออกจากโปรแกรม : 6\n");
     printf("\n\t\t\t          * เลือกโปรแกรมการทำงาน 1 - 6 *\n");
     ch1=getch();
     } while(ch1!='1'||ch1!='2'||ch1!='3'||ch1!='4'||ch1!='5'||ch1!='6');
    switch(ch1);
      {
     case'1':list_file();
      break;
     case'2':edit_file();
      break;
     case'3':delete_file();
      break;
     case'4':time_file();
      break;
     case'5':con_file();
      break;
     case'6':quit();
      break;
      }
    goto loop;
}
initdata()
{
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
 int n;
 FILE *fp1;
 FILE *fp2;
 for (n=0;n<100;++n)
    {
    data[n].name[0]        = '\0';
    data[n].sex[0]         = '\0';
    data[n].age            = '\0';
    data[n].salary         = '\0';
    data[n].address[0]     = '\0';
    data[n].position[0]    = '\0';
    data[n].birthday[0]    = '\0';
    data[n].nofbank[0]     = '\0';
    data[n].nofper[0]      = '\0';
    data[n].nofsocial[0]   = '\0';
    data[n].group[0]       = '\0';
    }
 if ((fp1=fopen("person.dat","rb"))==NULL)
    {
    puts("\n\tไม่สามารถเปิดแฟ้มข้อมูลได้\n");
    exit(0);
    }
 fread(&data,sizeof(data),1,fp1);
 fclose(fp1);
 if ((fp2=fopen("dbmain.cfg","rb"))==NULL)
    {
    puts("\n\tไม่สามารถเปิดแฟ้มข้อมูลได้\n");
    exit(0);
    }
 fread(&co,sizeof(co),1,fp2);
 fclose(fp2);
 return;
}
quit()
{
 FILE *fp1;
 if ((fp1=fopen("person.dat","wb"))==NULL)
    {
    puts("\n\tไม่สามารถเปิดแฟ้มข้อมูลได้\n");
    exit(0);
    }
 rewind(fp1);
 fwrite(&data,sizeof(data),1,fp1);
 fclose(fp1);
 exit(0);
}
con_file()
{
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
      char co;
      initco();
      loop:
      do {
      clrscr();
      printf("\n\n\n\n");
      printf("\t\t          ff  กำหนดค่าวันเวลาการทำงาน ff");
      printf("\n\n\n");
      printf("\t\t                    วันหยุดพิเศษ : 1");
      printf("\n");
      printf("\t\t              แก้ไขเวลาการทำงาน : 2");
      printf("\n");
      printf("\t\t             ออกจากโปรแกรม : 3");
      printf("\n\n");
      printf("\t\t      fff เลือกหมายเลขหัวข้อการทำงาน 1 - 3 fff");
      co = getch();
      }while(co!='1'&&co!='2'&&co!='3');
     switch(co)
        {
      case'1':daystop();
       break;
      case'2':otfile();
       break;
      case'3':return;
        }
        goto loop;
}
daystop()
{
 FILE *fp7;
 int k;
 char ch5,i;
 char spacday[13];
 char tmonth[12][11] = {"มกราคม","กุมภาพันธ์","มีนาคม","เมษายน"
                       ,"พฤษภาคม","มิถุนายน","กรกฎาคม","สิงหาคม"
                       ,"กันยายน","ตุลาคม","พฤศจิกายน","ธันวาคม" };
 struct date da;
 char styear[24][3] = { "37","38","39","40","41","42",
                        "43","44","45","46","47","48",
                        "49","50","51","52","53","54",
                        "55","56","57","58","59","60" };
 getdate(&da);
 k = da.da_year-1994      ;
 strcpy(spacday,"spday")  ;
 strcat(spacday,styear[k]);
 strcat(spacday,".lib")   ;
 if ((fp7=fopen(spacday,"rb"))==NULL)
       {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        puts("\n\n\n\n\t\t\t   ไม่สามารถเปิดแฟ้มข้อมูลได้\n");
        return;
        }
 clrscr();
 printf("\n\n\n");
 printf("\t\t              ๔๔ วันหยุดพิเศษ ๔๔\n");
 do
    {
    fread(&st,sizeof(2),1,fp7);
    i=st.stmonth-1;
    if(!feof(fp7)&&st.stdate!=0&&st.stdate<=31)
       {
       printf("\n\t\t\t     วันที่ %3d %s",st.stdate,tmonth[i]);
       }
    else printf("\n");
    }while(!feof(fp7)||st.stdate!=0||st.stdate<=31);
 fclose(fp7);
 printf("\n\n\t\tคุณต้องการเพิ่มเติมวันหยุดพิเศษใช่หรือไม่ ? (ใช่=1/ไม่ใช่=2).");
    ch5=(getchar());
    if(ch5=='2') return;
    else if(ch5=='1')
       {
       if ((fp7=fopen(spacday,"ab"))==NULL)
   {
   puts("\n\tไม่สามารถเปิดแฟ้มข้อมูลได้\n");
   return;
   }
       printf("\n\t\t ๔ กรุณาใส่เป็นหมายเลข ๔");
       printf("\n\t\t    วันที่     : ");scanf("%d",&st.stdate);
       printf("\n\t\t    เดือน    : ");scanf("%d",&st.stmonth);
       fwrite(&st,sizeof(2),1,fp7);
       fclose(fp7);
       }
    return;
}
initco()
{
 FILE *fp2;
 co[0].chi       = '\0';
 co[0].cmi       = '\0';
 co[0].cho       = '\0';
 co[0].cmo       = '\0';
 co[0].cothi     = '\0';
 co[0].cotmi     = '\0';
 co[0].cotho     = '\0';
 co[0].cotmo    = '\0';
 co[0].cotpn     = '\0';
 co[0].cotpsat   = '\0';
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
 co[0].cotpsun   = '\0';
 co[0].cotpsto   = '\0';
 co[0].cml       = '\0';
 co[0].cmlb      = '\0';
 if ((fp2=fopen("dbmain.cfg","r+t"))==NULL)
    {
    puts("\n\tไม่สามารถเปิดแฟ้มข้อมูลได้\n");
    exit(0);
    }
 fread(&co,sizeof(co),1,fp2);
 fclose(fp2);
 return;
}
otfile()
{
    FILE *fp2;
    char ch4;
    clrscr();
    printf("\n\n\n\t\t\t      กำหนดเวลาการทำงาน \n");
    printf("\n\t\t\t    เข้าทำงานเวลา%2d:%02d ",co[0].chi,co[0].cmi);
    printf("\n\t\t\t    ออกทำงานเวลา%3d:%02d ",co[0].cho,co[0].cmo);
    printf("\n\t\t\t    เวลาเข้าทำ OT%3d:%02d ",co[0].cothi,co[0].cotmi);
    printf("\n\t\t\t    เวลาออกทำ OT%3d:%02d ",co[0].cotho,co[0].cotmo);
    printf("\n\n\t\t    กำหนดการจ่ายเงิน OverTime (เป็นจำนวนเท่า)\n");
    printf("\n\t\t    OverTime ในวันธรรมดา      : %2.03f เท่า",co[0].cotpn);
    printf("\n\t\t    OverTime ในวันเสาร์       : %2.03f เท่า",co[0].cotpsat)
    printf("\n\t\t    OverTime ในวันอาทิตย์     : %2.03f เท่า",co[0].cotpsun)
    printf("\n\t\t    OverTime ในวันหยุดพิเศษ   : %2.03f เท่า",co[0].cotpsto)
    printf("\n\n\t\t      กำหนดเวลามาสายสูงสุด : %d นาที",co[0].cml);
    printf("\n\t กำหนดอัตราการหักเงินเดือน(เปอร์เซนต์/นาที) : %f เปอร์เซนต์/นาที",co[0
    printf("\n\n\t\tคุณต้องการแก้ไขค่า CONFIG ใช่หรือไม่ ? (ใช่=1/ไม่ใช่=2).");
    ch4=(getchar());
    if(ch4=='2') return;
    else if(ch4=='1')
       {
       if ((fp2=fopen("dbmain.cfg","w+t"))==NULL)
    {
    puts("\n\tไม่สามารถเปิดแฟ้มข้อมูลได้\n");
    exit(0);
    }
        printf("\n\t  ***        แก้ไขเป็น        ***\n\n");
        printf("\n\t            กำหนดเวลาการทำงาน ");
        printf("\n\t เข้าทำงานเวลา (ช.ม.) : ");
        scanf("%d",co[0].chi);
        printf("\t เข้าทำงานเวลา (นาที.) : ");
        scanf("%d",co[0].cmi);
        printf("\t ออกทำงานเวลา (ช.ม.) : ");
        scanf("%d",co[0].cho);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        printf("\t ออกทำงานเวลา (นาที.) : ");
        scanf("%d",co[0].cmo);
        printf("\t เข้าทำ OverTime (ช.ม.) : ");
        scanf("%d",co[0].cothi);
        printf("\t เข้าทำ OverTime (นาที.) : ");
        scanf("%d",co[0].cotmi);
        printf("\t ออกทำ OverTime (ช.ม.) : ");
        scanf("%d",co[0].cotho);
        printf("\t ออกทำ OverTime (นาที.) : ");
        scanf("%d",co[0].cotmo);
        printf("\n\n       กำหนดการจ่ายเงิน OverTime (เป็นจำนวนเท่า)");
        printf("\n\t OverTime ในวันธรรมดา      : ");
        scanf("%f",co[0].cotpn);
        printf("\t OverTime ในวันเสาร์        : ");
        scanf("%f",co[0].cotpsat);
        printf("\t OverTime ในวันอาทิตย์      : ");
        scanf("%f",co[0].cotpsun);
        printf("\t OverTime ในวันหยุดพิเศษ    : ");
        scanf("%f",co[0].cotpsto);
        printf("\n\n       กำหนดเวลามาสายสูงสุด(นาที) : ");
        scanf("%d",co[0].cml);
        printf("\n  กำหนดอัตราการหักเงินเดือน(%/นาที) : ");
        scanf("%f",co[0].cmlb);
        rewind(fp2);
        fwrite(co,sizeof(co),1,fp2);
        fclose(fp2);
        return;
        }
}
edit_file()
{
 char ch2,s[1];
 int n;
 printf("\tคุณต้องการแก้ไขข้อมูลพนักงานใช่หรือไม่ ? (ใช่=1/ไม่ใช่=2).");
 ch2=(getchar());
 if(ch2=='2')
    return;
 else if(ch2=='1')
    {
    printf("\tกรุณาใส่เลขรหัสที่คุณต้องการจะแก้ไข £ : ");
    scanf("%d",n);
    gets(s);
    printf("\n\t  ***       แก้ไขเป็น       ***\n\n");
    printf("    ชื่อ,นามสกุล : ");
    gets(data[n-1].name);
    printf("    เพศ (ชาย,หญิง) : ");
    gets(data[n-1].sex);
    printf("    ที่อยู่ : ");
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    gets(data[n-1].address);
    printf("    ตำแหน่ง : ");
    gets(data[n-1].position);
    printf("    วันเกิด : ");
    gets(data[n-1].birthday);
    printf("    หมายเลขบัญชีธนาคาร : ");
    gets(data[n-1].nofbank);
    printf("    หมายเลขบัตรประชาชน : ");
    gets(data[n-1].nofper);
    printf("    หมายเลขบัตรประกันสังคม : ");
    gets(data[n-1].nofsocial);
    printf("    กรุ๊บเลือด (A,B,AB,O) : ");
    gets(data[n-1].group);
    printf("    อายุ : ");
    scanf("%d",&data[n-1].age);
    printf("    เงินเดือน : ");
    scanf("%d",&data[n-1].salary);
    return;
    }
}
list_file()
{
 int n,m=0;
 for(n=0;n<100;n=n)
 {
    do
       {
    if(data[n].salary > 0)
       {
       printf("\n   รหัส : ff %2d ff   ชื่อ : %s\n",n+1,data[n].name);
       printf("   เพศ                 : %s\n",data[n].sex);
       printf("   ที่อยู่             : %s\n",data[n].address);
       printf("   ตำแหน่ง             : %s\n",data[n].position);
       printf("   วันเกิด             : %s\n",data[n].birthday);
       printf("   หมายเลขบัญชีธนาคาร  : %s\n",data[n].nofbank);
       printf("   หมายเลขบัตรประชาชน  : %s\n",data[n].nofper);
       printf("   หมายเลขบัตรประกันสังคม : %s\n",data[n].nofsocial);
       printf("   กรุ๊บโลอด           : %s\n",data[n].group);
       printf("   อายุ                : %d ปี\n",data[n].age);
       printf("   เงินเดือน           : %ld บาท\n",data[n].salary);
       m++;n++;
       }
    else
       {
       printf("\n      *** สิ้นสุดข้อมูลกรุณากดปุ่มใดๆ ***");
       getch();
       return;
       }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    }while(m!=2);
    getch();
    m=0;
 }
 printf("\tf == สิ้นสุดข้อมูล == f\n\t...กรุณากดปุ่มใดๆ...");
 getch();
 return;
}
delete_file()
{
 char ch3;
 int n;
 printf("\tคุณต้องการลบข้อมูลพนักงานใช่หรือไม่ ? (ใช่=1/ไม่ใช่=2).");
 ch3=(getchar());
 if (ch3=='2')return;
 else if (ch3=='1')
    {
    printf("\tกรุณาใส่รหัสที่คุณต้องการลบออกจากแฟ้มข้อมูล f : ");
    scanf("%d",&n);
    data[n-1].name[0]    ='\0';
    data[n-1].sex[0]     ='\0';
       data[n-1].age        ='\0';
    data[n-1].salary     ='\0';
    printf("\n\tข้อมูลรหัสที่ f %d ได้ถูกลบออกจากไฟล์ข้อมูลแล้ว.\n",n);
    printf("\t                ...กรุณากดปุ่มใดๆ...");
    getch();
    return;
    }
}
time_file()
{
     char mo;
     loop:
     do {
     clrscr();
     printf("\n\n\n\n");
     printf("\t\t    ต้องการดูเวลาการทำงานของพนักงานในเดือนใด ?");
     printf("\n\n\n");
     printf("\t\t        ดูเวลาการทำงานในเดือนที่ผ่านๆมา : 1");
     printf("\n");
     printf("\t\t              ดูเวลาการทำงานในเดือนนี้ : 2");
     printf("\n");
     printf("\t\t                 ออกจากโปรแกรม : 3");
     printf("\n\n");
     printf("\t\t    fff เลือกหมายเลขหัวข้อการทำงาน 1 - 3 fff");
     mo = getch();
     }while(mo!='1'&&mo!='2'&&mo!='3');
    switch(mo)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        {
      case'1':oldmonth();
       break;
      case'2':thismonth();
       break;
      case'3':return;
        }
       goto loop;
}
oldmonth()
{
     FILE *fp3;
     char fmonth[9];
     clrscr();
     if ((fp3=fopen("month.lib","r"))==NULL)
    {
    puts("\n\n\n\n\t  ไม่สามารถเปิดแฟ้มข้อมูล month.lib ได้\n");
    printf("\t          กรุณาตรวจสอบแฟ้มข้อมูล");
    return;
    }
     printf("\n\n\n");
     printf("\t\t     เดือนที่มีอยู่ในแฟ้มข้อมูล month.lib\n\n");
     do
 {
 fread(mon,sizeof(mon),1,fp3);
 if (feof(fp3))
    {
    printf(" %s ",&mon.mon);
    }
 else printf("\n");
 }while(!feof(fp3));
     fclose(fp3);
     printf("\n\n\t    กรุณาใส่ชื่อไฟล์ในเดือนที่ต้องการ : ");
     scanf("%s",fmonth);
     mtime(&fmonth);
     return;
}
thismonth()
{
      FILE *fp4;
      FILE *fp5;
      char snum;
      int  num;
      struct date da;
      char strmon[12][4] = { "jan","feb","mar","apr","may","jun",
                             "jul","aug","sep","oct","nov","dec" };
      char stryear[24][3] = { "37","38","39","40","41","42",
                              "43","44","45","46","47","48",
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                                "49","50","51","52","53","54",
                                "55","56","57","58","59","60" };
        char fmonth[9]; char fmonth2[9];
        int i,j;
        getdate(da);
        i = da.da_mon-1; j = da.da_year-1994;
        strcpy(fmonth2,strmon[i]) ; strcpy(fmonth,strmon[i]);
        strcat(fmonth2,stryear[j]); strcat(fmonth,stryear[j]);
        strcat(fmonth2,".asc")    ; strcat(fmonth,".dat");
        clrscr();
        if ((fp4 = fopen(fmonth2,"r"))==NULL)
  {
  printf("\n\t    ไฟสามารถเปิดแฟ้มข้อมูล %s นี้ได้\n",&fmonth2);
  getch();
  return;
  }
        if ((fp5 = fopen(fmonth,"ab"))==NULL)
  {
  printf("\n\t    ไฟสามารถเปิดแฟ้มข้อมูล %s นี้ได้\n",&fmonth);
  getch();
  fclose(fp4);
  return;
  }
        do {
  fgets(snum,3,fp4);
  num=0;
  num = num+((snum[0]-'0')*10)+(snum[1]-'0');
  fputc(num,fp5);
  snum[0]='0';snum[1]='0';
  } while(feof(fp4));
        fclose(fp4);
        fclose(fp5);
        mtime(&fmonth);
        remove(fmonth);
        return;
}
mtime(fmonth)
char fmonth[9];
{
     struct date da;
     FILE *fp6,*fp7;
     long rsalary=0;
     int k,n,milate=0,mmilate=0,bu=0,otnorm,otnorn;
     int wotnor=0,wotsat=0,wotsun=0,wotsto=0;
     char nmon,j,o=0;
     char namem[11],spacday[13],dwmonth[13];
     char day [7][8]={"จันทร์","อังคาร","พุธ",
                      "พฤหัส","ศุกร์","เสาร์","อาทิตย์"};
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    char styear[24][2] = { "37","38","39","40","41","42",
                           "43","44","45","46","47","48",
                           "49","50","51","52","53","54",
                           "55","56","57","58","59","60" };
    getdate(&da);
    k = da.da_year-1994      ;
    namem[0]='\0';
    otnorm=((co[0].cotho-co[0].cothi)*60)+(co[0].cotmo-co[0].cotmi);
    if (fmonth[0]=='j'&&fmonth[1]=='a'&&fmonth[2]=='n'){
nmon=1;strcat(namem,"มกราคม");}
    else if (fmonth[0]=='f'&&fmonth[1]=='e'&&fmonth[2]=='b'){
nmon=2;strcat(namem,"กุมภาพันธ์");}
    else if (fmonth[0]=='m'&&fmonth[1]=='a'&&fmonth[2]=='r'){
nmon=3;strcat(namem,"มีนาคม");}
    else if (fmonth[0]=='a'&&fmonth[1]=='p'&&fmonth[2]=='r'){
nmon=4;strcat(namem,"เมษายน");}
    else if (fmonth[0]=='m'&&fmonth[1]=='a'&&fmonth[2]=='y'){
nmon=5;strcat(namem,"พฤษภาคม");}
    else if (fmonth[0]=='j'&&fmonth[1]=='u'&&fmonth[2]=='n'){
nmon=6;strcat(namem,"มิถุนายน");}
    else if (fmonth[0]=='j'&&fmonth[1]=='u'&&fmonth[2]=='l'){
nmon=7;strcat(namem,"กรกฎาคม");}
    else if (fmonth[0]=='a'&&fmonth[1]=='u'&&fmonth[2]=='g'){
nmon=8;strcat(namem,"สิงหาคม");}
    else if (fmonth[0]=='s'&&fmonth[1]=='e'&&fmonth[2]=='p'){
nmon=9;strcat(namem,"กันยายน");}
    else if (fmonth[0]=='o'&&fmonth[1]=='c'&&fmonth[2]=='t'){
nmon=10;strcat(namem,"ตุลาคม");}
    else if (fmonth[0]=='n'&&fmonth[1]=='o'&&fmonth[2]=='v'){
nmon=11;strcat(namem,"พฤศจิกายน");}
    else if (fmonth[0]=='d'&&fmonth[1]=='e'&&fmonth[2]=='c'){
nmon=12;strcat(namem,"ธันวาคม");}
    else return;
    strcpy(spacday,"spday")  ;
    strcat(spacday,styear[k]);
    strcat(spacday,".lib")   ;
    if ((fp7=fopen(spacday,"rb"))==NULL)
{
puts("\n\n\n\n\t\t  ไม่สามารถเปิดแฟ้มข้อมูล spday??.lib ได้\n");
getch();
return;
}
    else
{
if ((fp6=fopen(fmonth,"rb"))==NULL)
   {
   printf("\n\n\t          ไม่สามารถเปิดไฟล์ %s ได้",fmonth);
   printf("\n\t              กรุณาตรวจสอบแฟ้มข้อมูล\n");
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    getch();
    return;
    }
else
    {
    printf("\n\t    กรุณาใส่รหัสพนักงานที่ต้องการดูเวลาการทำงาน : ");
    scanf("%d",cod2);
    printf("\n\t\t\t    เดือน %s\n",&namem);
    n = cod2;
    if(n<=100&&n!=0)
       {
       printf("\t\t\tนาย %s \n\n",data[n-1].name);
       }
    else
       {
       printf("\n\n\n\t\t\t  ff กรุณาใส่รหัสพนักงานให้ถูกต้อง ff");
       getch();
       return;
       }
    do
       {
       do
  {
  fread(ti,sizeof(ti),1,fp6);
  if (ti.code == cod2)
    {
    j=ti.day-1;
    printf("  วัน   %7s  ที่%4d เข้าเวลา %2d:%02d ออกเวลา %2d:%02d *",
    day[j],ti.date,ti.hourin,ti.minutin,ti.hourout,ti.minutout);
    o++;
    bu=((ti.hourin-co[0].chi)*60)+(ti.minutin-co[0].cmi);
    if(bu > 0)
{
milate=milate+bu;
printf(" สาย *");
}
    else;
    bu=0;
    bu=((co[0].cho-ti.hourout)*60)+(co[0].cmo-ti.minutout);
    if(bu > 0)
{
printf(" ออกก่อน *");
}
    else;
    do{
fread(&st,sizeof(st),1,fp7);
if(ti.date==st.stdate&&nmon==st.stmonth)
       {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
      printf(" OverTime *");
      bu=0;
      bu=((ti.hourin-co[0].chi)*60)+(ti.minutin-co[0].cmi);
      if(bu > 0)/*ถ้าเข้างานก่อนเวลาที่กำหนดให้ใช้เวลาที่กำหนด*/
  {
wotsto=wotsto+(((ti.hourout-ti.hourin)*60)+(ti.minutout-ti.minutin)-60);
  }
      else
  {
wotsto=wotsto+(((ti.hourout-co[0].chi)*60)+(ti.minutout-co[0].cmi)-60);

  }
      }
      }while(!feof(fp7));
    if(st.stdate!=ti.date&&st.stmonth!=nmon)
      {
   if(ti.day==6)/*ถ้าเป็นวันเสาร์*/
      {
      printf(" OverTime *");
      bu=0;
      bu=((ti.hourin-co[0].chi)*60)+(ti.minutin-co[0].cmi);
      if(bu > 0)
  {
wotsat=wotsat+(((ti.hourout-ti.hourin)*60)+(ti.minutout-ti.minutin)-60);
  }
      else
  {
wotsat=wotsat+(((ti.hourout-co[0].chi)*60)+(ti.minutout-co[0].cmi)-60);
  }
      }
   else if(ti.day==7)/*ถ้าเป็นวันอาทิตย์*/
      {
      printf(" OverTime *");
      bu=0;
            bu=((ti.hourin-co[0].chi)*60)+(ti.minutin-co[0].cmi);
      if(bu > 0)
  {
wotsun=wotsun+(((ti.hourout-ti.hourin)*60)+(ti.minutout-ti.minutin)-60);
  }
      else
  {
wotsun=wotsun+(((ti.hourout-co[0].chi)*60)+(ti.minutout-co[0].cmi)-60);
  }
      }
    else if(ti.day<=5&&ti.day>=1)/*ถ้าเป็นวันจันทร์-ศุกร์*/
      {
      bu=0;
      bu=((ti.hourout-co[0].cothi)*60)+(ti.minutout-co[0].cotmi);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
      if(bu > 0)
  {
  printf(" OverTime *");
  bu=0;
   bu=((ti.hourout-co[0].cotho)*60)+(ti.minutout-co[0].cotmo);
   if(bu>0&&bu==0)
      {
      wotnor=wotnor+otnorm;
      }
   else
      {
otnorn=((ti.hourout-co[0].cothi)*60)+(ti.minutout-co[0].cotmi);
wotnor=wotnor+otnorn;
      }
  }
      else;
      }
   else printf("\n  ff ข้อมูลผิดพลาดกรุณาตรวจสอบข้อมูล ff");
}
printf("\n");
rewind(fp7);
}
}while(feof(fp6)&&o!=1);
getch();
o=0;
}while(!feof(fp6));
fclose(fp6);
if (ti.code != cod2)
   {
   mmilate=(milate-co[0].cml);
   if(mmilate>0)
      {
    rsalary=data[n-1].salary-(mmilate*co[0].cmlb*data[n-1].salary);
      }
   else
      {
      rsalary=data[n-1].salary;
      }
    rsalary=rsalary+((wotnor*co[0].cotpn*data[n-1].salary)/14400);
    rsalary=rsalary+((wotsat*co[0].cotpsat*data[n-1].salary)/14400);
    rsalary=rsalary+((wotsun*co[0].cotpsun*data[n-1].salary)/14400);
    rsalary=rsalary+((wotsto*co[0].cotpsto*data[n-1].salary)/14400);
   fclose(fp7);
   clrscr();
   printf("\n\n\n");
   printf("\t\t\t          เดือน %s",&namem);
   printf("\n\n\t\t\t        นาย %s ",data[n-1].name);
   printf("\n\n\t\t\t            สาย %2d นาที \n",milate);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    printf("\n\t\t          ทำ OT ในวันธรรมดา      %d นาที ",wotnor);
    printf("\n\t\t          ทำ OT ในวันเสาร์       %d นาที ",wotsat);
    printf("\n\t\t          ทำ OT ในวันอาทิตย์     %d นาที ",wotsun);
    printf("\n\t\t          ทำ OT ในวันหยุดพิเศษ   %d นาที ",wotsto);
    printf("\n\n\n\t\t        $$ เงินเดือนในเดือนนี้ %ld บาท $$",rsalary);
    getch();
    fclose(fp6);
    }
 return;
 }
    }
}
```


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
   fff Program CHECK1.C FOR AUTOMATIC SALARY CALCULATING fff

finclude <dos.h>
finclude <stdio.h>
finclude <time.h>
struct date da;
struct ptime
 {
 char code;
 char day;
 char date;
 char hourin;
 char minutin;
 char hourout;
 char minutout;
 } ti;
main()
{
 getdate(&da);
 if(da.da_day==1)
    {
    if(da.da_mon==1)
       {
       day1m1();
       }
    else
       {
       day1();
       }
    }
 else exit(0);
}
day1()
{
 FILE *fp1,*fp2,*fp3;
 int i,j;
 char num;
 char fmonth1[10],fmonth2[10],snum[2];
 char strmon[11][4] = { "jan","feb","mar","apr","may","jun",
                        "jul","aug","sep","oct","nov" };
 char stryear[24][3] = { "37","38","39","40","41","42",
                         "43","44","45","46","47","48",
                         "49","50","51","52","53","54",
                         "55","56","57","58","59","60" };
 getdate(&da);
 i = da.da_mon-2; j = da.da_year-1994;
 strcpy(fmonth1,strmon[i]) ; strcpy(fmonth2,strmon[i]) ;
 strcat(fmonth1,stryear[j]); strcat(fmonth2,stryear[j]);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
strcat(fmonth1,".asc")    ; strcat(fmonth2,".dat")    ;
clrscr();
   if ((fp1 = fopen(fmonth1,"r"))==NULL)
      {
      printf("\n\t   ไม่สามารถเปิดแฟ้มข้อมูล %s นี้ได้\n",&fmonth1);
      getch();
      exit(0);
      }
   if ((fp2 = fopen(fmonth2,"ab"))==NULL)
      {
      printf("\n\t   ไม่สามารถเปิดแฟ้มข้อมูล %s นี้ได้\n",&fmonth2);
      getch();
      fclose(fp1);
      exit(0);
      }
   do
      {
      fgets(snum,3,fp1);
      num = ((snum[0]-'0')*10)+(snum[1]-'0');
      fputc(num,fp2);
      snum[0]='0';snum[1]='0';
      } while(!feof(fp1));
   fclose(fp1);fclose(fp2);
   if ((fp3=fopen("month.lib","a"))==NULL)
      {
      puts("\n\n\n\n\t  ไม่สามารถเปิดแฟ้มข้อมูล month.lib ได้\n");
      printf("\t        กรุณาตรวจสอบแฟ้มข้อมูล");
      getch();
      exit(0);
      }
   fwrite(&fmonth2,sizeof(fmonth2),1,fp3);
   fclose(fp3);
}
day1m1()
{
 FILE *fp1,*fp2,*fp3;
 int k;
 char num;
 char fmonth1[10],fmonth2[10],snum[2];
 char stry[24][3] = { "37","38","39","40","41","42",
        "43","44","45","46","47","48",
        "49","50","51","52","53","54",
        "55","56","57","58","59","60" };
 getdate(&da);
 k = da.da_year-1995;
 strcpy(fmonth1,"dec")    ; strcpy(fmonth2,"dec")       ;
 strcat(fmonth1,stry[k]) ; strcat(fmonth2,stry[k])     ;
 strcat(fmonth1,".asc")   ; strcat(fmonth2,".dat")     ;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

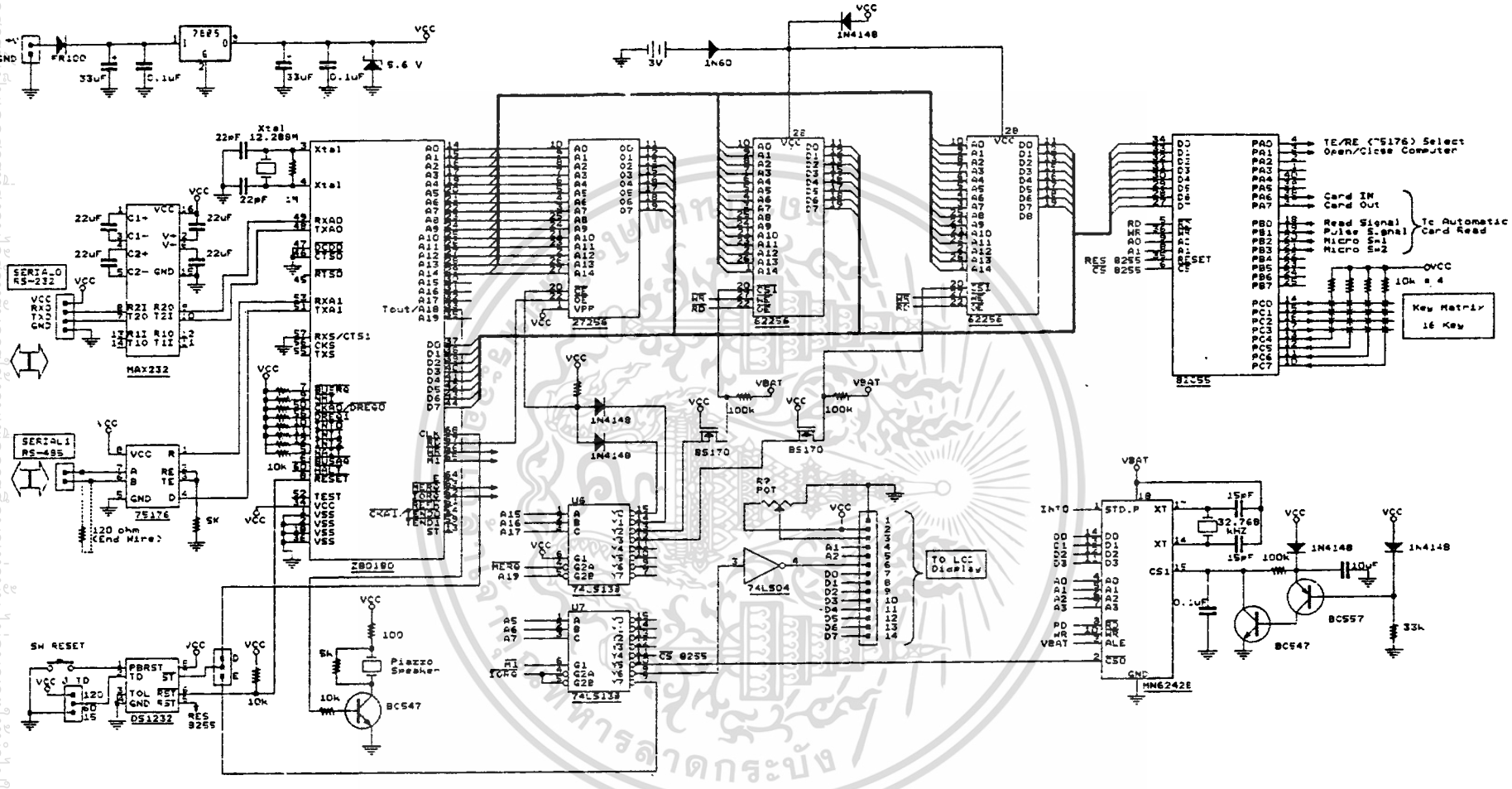
```
clrscr();
   if ((fp1 = fopen(fmonth1,"r"))==NULL)
      {
      printf("\n\t    ไม่สามารถเปิดแฟ้มข้อมูล %s นี้ได้\n",&fmonth1);
      getch();
      exit(0);
      }
   if ((fp2 = fopen(fmonth2,"ab"))==NULL)
      {
      printf("\n\t    ไม่สามารถเปิดแฟ้มข้อมูล %s นี้ได้\n",&fmonth2);
      getch();
      fclose(fp1);
      exit(0);
      }
   do
      {
      fgets(snum,3,fp1);
      num = ((snum[0]-'0')*10)+(snum[1]-'0');
      fputc(num,fp2);
      snum[0]='0';snum[1]='0';
      } while(!feof(fp1));
   fclose(fp1);fclose(fp2);
   if ((fp3=fopen("month.lib","a"))==NULL)
      {
      puts("\n\n\n\n\t  ไม่สามารถเปิดแฟ้มข้อมูล month.lib ได้\n");
      printf("\t          กรุณาตรวจสอบแฟ้มข้อมูล");
      return;
      }
   fwrite(&fmonth2,sizeof(fmonth2),1,fp3);
   fclose(fp3);
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
   £££ Program LBUFFER.C FOR AUTOMATIC SALARY CALCULATING £££

£include <dos.h>
£include <stdio.h>
£include <time.h>
struct date da;
main()
{
 FILE *fp1,*fp2;
 int i,j;
 char snum;
 char fmonth1[11];char fmonth2[10];
 char strmon[12][4] = { "jan","feb","mar","apr","may","jun",
                        "jul","aug","sep","oct","nov","dec" };
 char stryear[24][3] = { "37","38","39","40","41","42",
                         "43","44","45","46","47","48",
                         "49","50","51","52","53","54",
                         "55","56","57","58","59","60" };
 getdate(&da);
 i = da.da_mon-1; j = da.da_year-1994;
 strcpy(fmonth2,strmon[i]) ;
 strcat(fmonth2,stryear[j]);
 strcat(fmonth2,".asc")    ;
 strcpy(fmonth1,"buffer.asc");
 clrscr();
    if ((fp1 = fopen(fmonth1,"r"))==NULL)
       {
       printf("\n\t   ไม่สามารถเปิดแฟ้มข้อมูล buffer.asc นี้ได้\n");
       getch(); exit(0);
       }
    if ((fp2 = fopen(fmonth2,"ab"))==NULL)
       {
       printf("\n\t   ไม่สามารถเปิดแฟ้มข้อมูล %s นี้ได้\n",&fmonth2);
       getch();
       fclose(fp1); exit(0);
       }
     do
       {
            snum=fgetc(fp1);
       if (!feof(fp1))
    {
    fputc(snum,fp2);
    }
       else exit(0);
       } while(!feof(fp1));
     fclose(fp1);fclose(fp2);
     exit(0);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

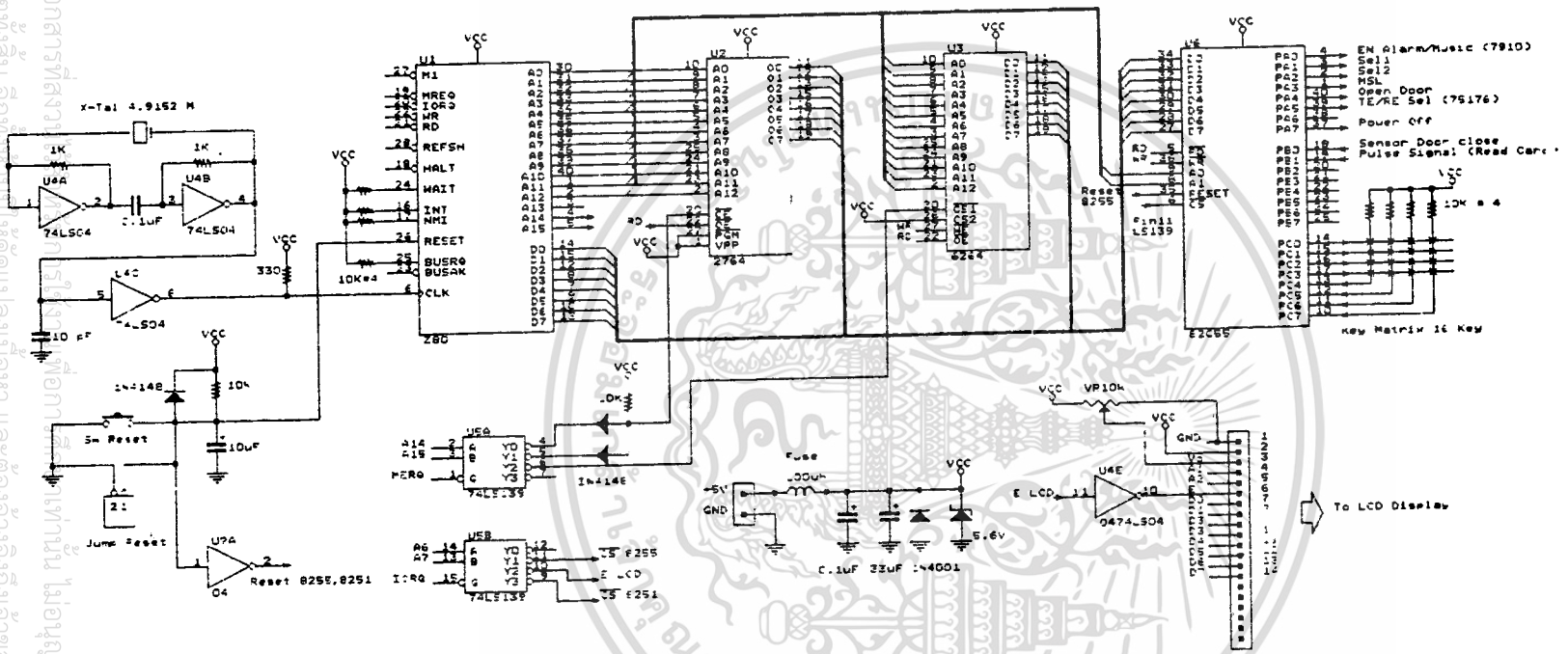


Center Module

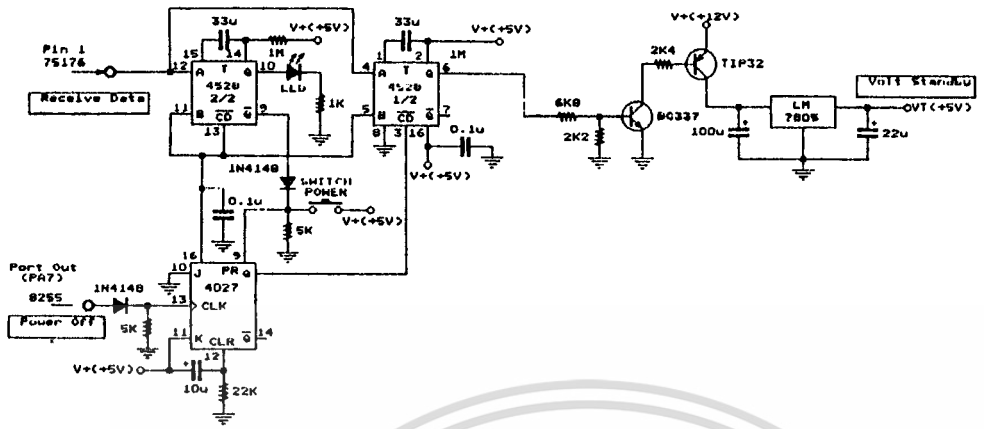
ภาคผนวก ข

วงจร และตัวเครื่อง

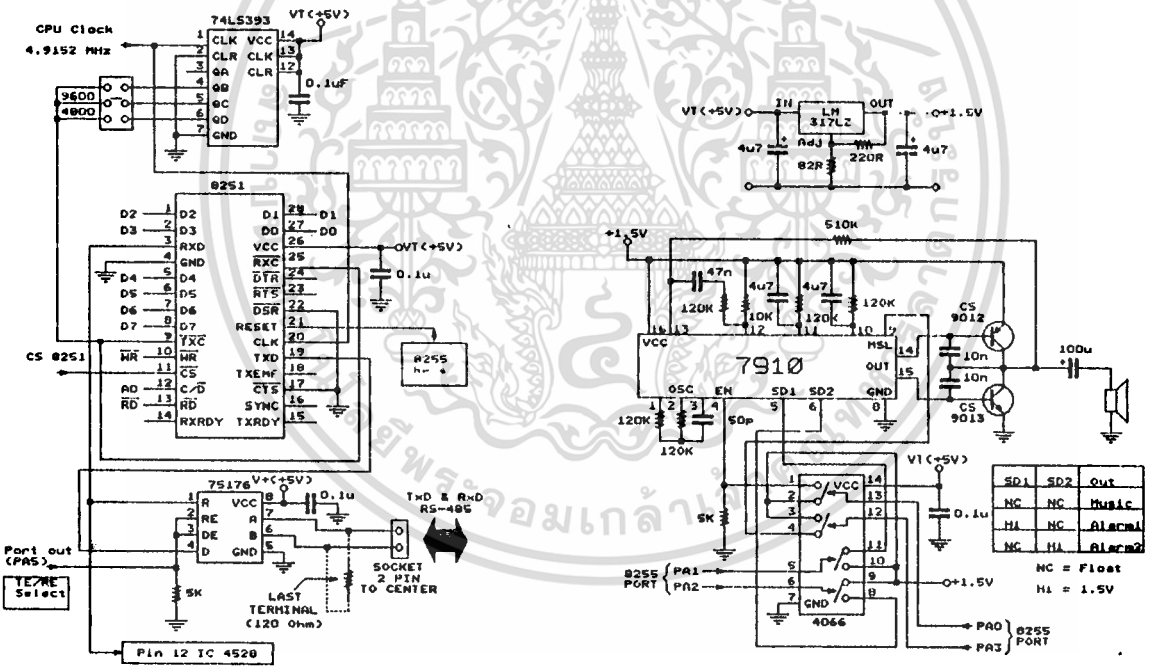
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Terminal Circuit



Standby Circuit

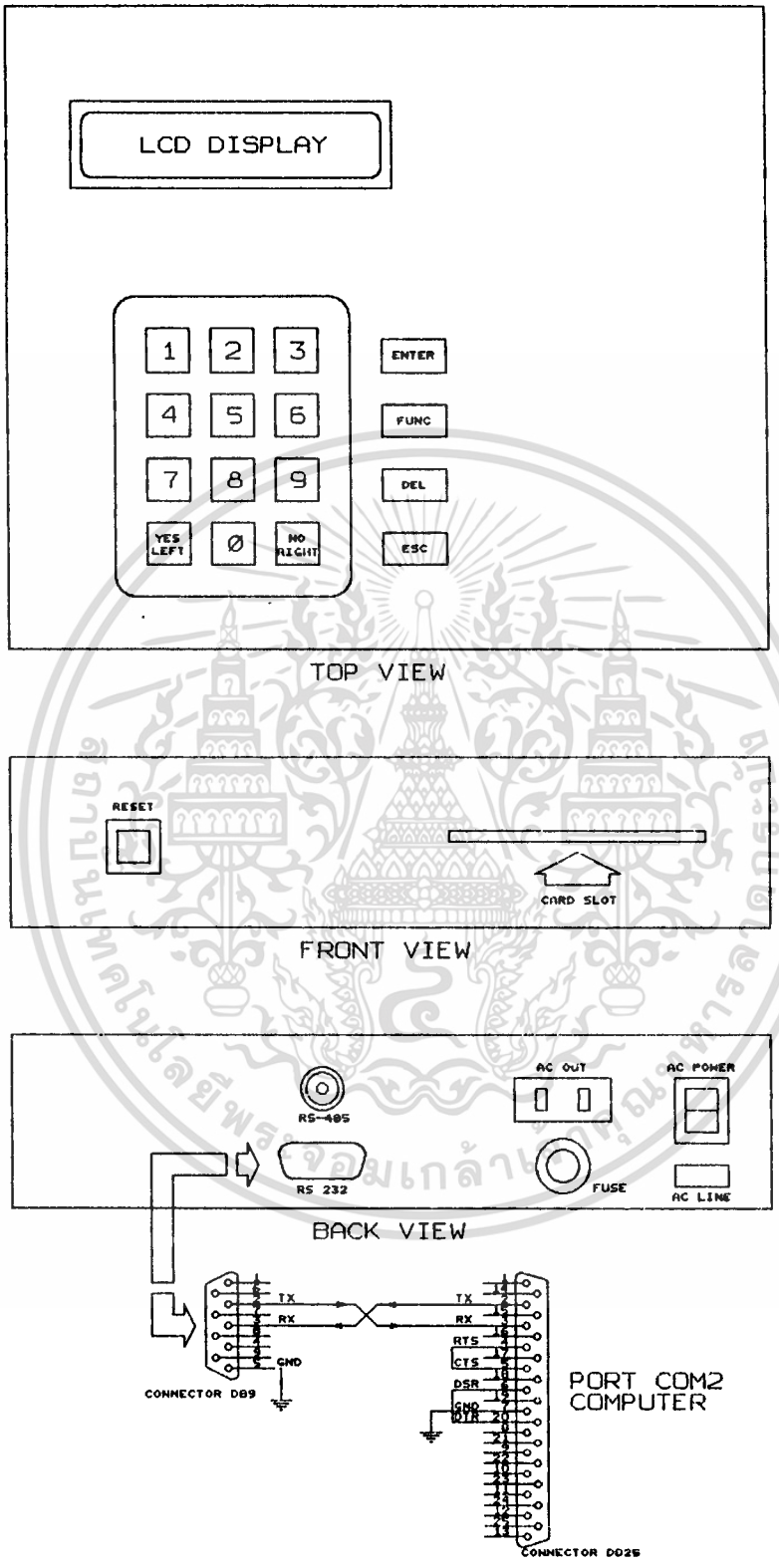


Serial Data Circuit

Music / Alarm Circuit

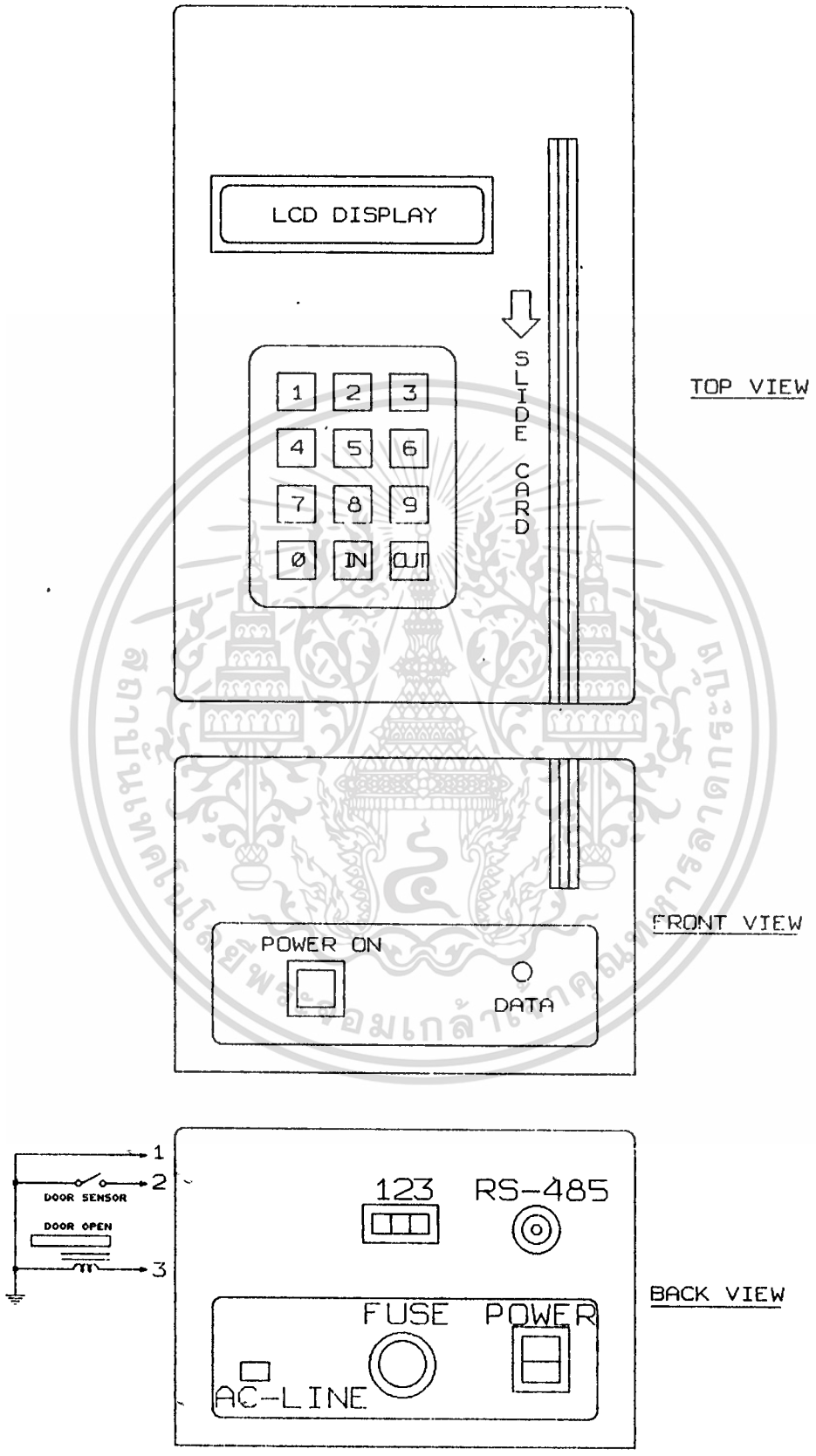
Terminal Module

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปแสดงตัวเครื่อง Center

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป แสดงตัวเครื่อง Terminal

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้