การยกระดับความปลอดภัยในการส่งข้อมูลโดยใช้สมาร์ตการ์ด

To Enhance Transmission Security by Using Smartcard

โดย

นาย คีริศวร์ นิธิพิพิธชัย

นาย อุกฤษ คลังวิฑูรย์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมสารสนเทศ

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2547

TO ENHANCE TRANSMISSION SECURITY BY USING SMARTCARD

BY

MR. KEERIS NITHIPIPITCHAI

MR. UKRIT KLANGVITOON

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF

THE REQUIREMENT FOR THE DEGREE OF

BACHELOR IN DEPARTMENT OF INFORMATION ENGINEERING

FACULTY OF ENGINEERING

KING MONGKUT'S INSTITUE OF TECHNOLOGY LARDKRABANG

2004

หัวข้อปริญญานิพนธ์	การยกระดับความปลอดภัยในการส่งข้อมูลโดยใช้สมาร์ตการ์ด	
ชื่อนักศึกษา	นาย คีริศวร์ นิธิพิพิธชัย	รหัสประจำตัว 44010616
	นาย อุกฤษ คลังวิฑูรย์	รหัสประจำตัว 44010617
อาจารย์ที่ปรึกษา	ผศ. ไพศาล สิทธิโยภาสกุล	
	รศ.ดร. ปีติเขต สู้รักษา	
ระดับการศึกษา	ปริญญาตรี วิศวกรรมศาสตรบัณฑิต	
	สาขาวิศวกรรมสารสนเทศ	
ภาควิชา	วิศวกรรมสารสนเทศ	
ปีการศึกษา	2547	

ปริญญานิพนธ์ฉบับนี้ได้รับความเห็นชอบจากอาจารย์ที่ปรึกษาเป็นที่เรียบร้อยแล้ว

.....

(ผศ. ไพศาล สิทธิโยภาสกุล)

อาจารย์ผู้ควบคุมวิทยานิพนธ์

.....

(รศ.ดร. ปีติเขต สู้รักษา)

อาจารย์ผู้ควบคุมวิทยานิพนธ์

หัวข้อปริญญานิพนธ์	การยกระดับความปลอดภัยในการส่งข้อมูลโดยใช้สมาร์ตการ์ด	
ชื่อนักศึกษา	นาย คีริศวร์ นิธิพิพิธชัย	รหัสประจำตัว 44010616
	นาย อุกฤษ คลังวิฑูรย์	รหัสประจำตัว 44010617
อาจารย์ที่ปรึกษา	ผศ. ไพศาล สิทธิโยภาสกุล	
	รศ.ดร. ปีลิเขต สู้รักษา	
ระดับการศึกษา	ปริญญาตรี วิศวกรรมศาสตรบัณฑิต	
	สาขาวิศวกรรมสารสนเทศ	
ภาควิชา	วิศวกรรมสารสนเทศ	
ปีการศึกษา	2547	

บทคัดย่อ

โครงงานนี้ มุ่งเน้นไปในด้านการรักษาความปลอดภัยของการส่งข้อมูลผ่านระบบเครือข่าย โดยการนำเอาการเข้ารหัสลับข้อมูลมาประยุกต์ใช้ร่วมกับสมาร์ตการ์ด หลักการทำงานคือ ภายในสมาร์ตการ์ดจะบันทึกกุญแจรหัส ที่ใช้ในการเข้ารหัสลับและถอดรหัสลับอยู่เป็นจำนวนมาก พร้อมทั้งมีลำดับของกุญแจรหัสกำกับอยู่ เมื่อต้องการส่งข้อมูล ขั้นตอนแรกคือการเสียบบัตรสมาร์ตการ์ดเข้าที่เครื่องอ่านที่เชื่อมต่ออยู่กับพอร์ทคอมพิวเตอร์ แล้วเปิดโปรแกรมที่ใช้ในการเข้ารหัสลับขึ้นมา ซึ่งข้อมูลจะถูกเข้ารหัสลับโดยกุญแจรหัสที่ถูกสุ่มมาจากชุดของกุญแจรหัสภายในสมาร์ตการ์ด จากนั้นข้อมูลที่เข้ารหัสลับแล้วจะถูกส่งผ่านเครือข่ายพร้อมกับลำดับของกุญแจรหัสที่ใช้ เมื่อถึงฝ่ายผู้รับก็จะเสียบบัตรสมาร์ตการ์ดซึ่งมีข้อมูลของกุญแจรหัสภายในตรงกันกับทางด้านของผู้ส่ง และใช้โปรแกรมในการถอดรหัสลับ โดยใช้กุญแจรหัสที่ตรงกับลำดับของกุญแจรหัสที่ถูกส่งมา เพราะฉะนั้นถึงแม้ว่าจะมีการดักจับข้อมูลระหว่างทางก็จะไม่สามารถอ่านข้อมูลที่ถูกต้องได้ เนื่องจากข้อมูลที่ได้ไปจะเป็นเพียงข้อมูลที่ถูกเข้ารหัสลับแล้วกับลำดับของกุญแจรหัสเท่านั้น

Thesis Title	To enhance transmission security by using smartcard	
Student	Mr. Keeris Nithipipitchai	ID. 44010616
	Mr. Ukrit Klangvitoon	ID. 44010617
Advisor	Asst. Prof. Paisarn Sidthiyopasakul	
	Assoc. Prof. Dr. Pitikhate Sooraksa	
Graduate Level	Bachelor Degree of Information Engineering	
Department	Information Engineering	
Academic Year	2004	

Abstract

This project has been constantly generated with an intention to study about security of the network information transmission by applying the encryption process together with the smart card. In other words, the smart card will record the numbers of secret keys that are entirely required for encryption and decryption processes, as well as provided secret keys order.

To transfer the information, the first step is to insert the smart card into the smart card reader is connected to computer port; then the next process will be conducted via the encryption program. The information will be encrypted by secret key which has been randomly chosen from group of secret key from the smart card, after that all the encrypted information will be sent through the network along with the secret key order that has been used.

Then the receiver will insert the smart card which contains the secret key information that matches with the one of the sender. The decryption process will be conducted by using the secret key order that was provided. Therefore, even there is any attempt to sniff the information during the transmission, the information would not be read correctly as the stolen information will only be the encrypted information and the secret key order.

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้คงไม่อาจสำเร็จได้ หากไม่มี ผศ. ไพศาล สิทธิโยภาสกุล และ รศ.ดร. ปีติเขต สู้รักษา อาจารย์ที่ปรึกษาปริญญานิพนธ์ ผู้ให้คำปรึกษา แนะนำ ตรวจแก้ไข และเอาใจใส่ตลอดระยะเวลาทั้งหมดที่ทำปริญญานิพนธ์ ซึ่งคณะผู้จัดทำรู้สึกซาบซึ้งและขอกราบขอบพระคุณอาจารย์เป็นอย่างยิ่ง

ขอขอบคุณอาจารย์กฤดากร กล่อมการ ที่ช่วยให้คำแนะนำและตอบข้อซักถามของพวกเราอย่างเต็มใจ รวมทั้งเพื่อนๆในภาควิชาวิศวกรรมสารสนเทศทุกคน สำหรับมิตรภาพและน้ำใจที่มีให้กันเสมอมา

สุดท้ายนี้คณะผู้จัดทำต้องขอกราบขอบพระคุณ บุคคลที่สำคัญที่สุดที่ทำให้มีวันนี้ นั่นคือ บิดา มารดา ที่เคารพรักของคณะผู้จัดทำ สำหรับความห่วงใย กำลังใจ ความเข้าใจ และคอยสนับสนุนในทุกๆด้านตลอดมา

สารบัญ

สารบัญ (ต่อ)

สารบัญ (ต่อ)

สารบัญรูปภาพ

สารบัญรูปภาพ (ต่อ)

สารบัญรูปภาพ (ต่อ)

สารบัญตาราง

บทที่ 1
บทนำ

1.1 แนวคิดเริ่มต้นในการทำโครงงาน

ในปัจจุบันนี้ ข้อมูลข่าวสารเป็นสิ่งที่มีความสำคัญและจำเป็นอย่างมากในทุกๆด้าน ทั้งทางด้านการเมือง เศรษฐกิจ สังคม ฯลฯ บุคคลที่มีข้อมูลที่รวดเร็ว ถูกต้อง ย่อมมีความได้เปรียบมากกว่าบุคคลอื่น และในอีกแง่มุมหนึ่ง บุคคลที่สามารถลักลอบเพื่อดูข้อมูล หรือเข้าไปทำลายข้อมูลของศัตรูหรือคู่แข่งนั้น ก็สามารถก่อให้เกิดความเสียหายได้อย่างมหาศาลเช่นกัน ดังนั้นการป้องกันและการรักษาความปลอดภัยของข้อมูลข่าวสารจึงเป็นสิ่งที่ควรให้ความสำคัญอย่างหลีกเลี่ยงไม่ได้เพื่อป้องกันความเสียหายที่อาจจะเกิดขึ้น

การป้องกันและรักษาความปลอดภัยในปัจจุบันนั้นมีอยู่มากมายหลายระบบหลายวิธี ทั้งการใช้อุปกรณ์ที่เป็นฮาร์ดแวร์ เช่น ไฟร์วอล VPN หรือการใช้ซอฟท์แวร์ เช่น การใช้โปรแกรมเข้ารหัสลับข้อมูล โปรแกรมไฟร์วอล เป็นต้น ซึ่งไม่ว่าจะเป็นการป้องกันแบบไหน ก็มักจะมีช่องโหว่อยู่เสมอ ดังนั้นทางคณะผู้จัดทำจึงได้มีแนวความคิดที่จะพัฒนาและสร้างระบบรักษาความปลอดภัยของการส่งข้อมูลผ่านทางเครือข่ายลักษณะใหม่ขึ้นมา เพื่อให้เกิดเป็นแนวทางเลือกใหม่ในการใช้งาน และสามารถนำไปประยุกต์ใช้กับระบบการป้องกันอื่นๆ เพื่อให้การรักษาความปลอดภัยของข้อมูลให้มีประสิทธิภาพมากยิ่งขึ้น

ในการศึกษาวิชา Internet System ชั้นปีที่ 3 จากบทเรียนเรื่อง Protocol และ Security ที่กล่าวถึงการสนิฟเฟอร์ดักจับข้อมูลและกุญแจรหัสที่ใช้ในการเข้ารหัสลับข้อมูล ทำให้คณะผู้จัดทำเกิดแนวคิดที่จะกำจัดข้อบกพร่องบางส่วนออกไป และพัฒนาทำให้การเข้ารหัสลับข้อมูลมีความซับซ้อนและยากต่อการลักลอบหรือการโจมตีมากขึ้น โดยการใช้การเข้ารหัสลับข้อมูลที่ใช้กุญแจรหัสในการเข้ารหัสลับจากอุปกรณ์ภายนอก คือ สมาร์ตการ์ด ซึ่งเป็นอุปกรณ์ที่มีหน่วยความจำในตัวเอง สามารถเก็บข้อมูลไว้ภายในได้ มีแนวโน้มที่จะพัฒนาและมีการใช้อย่างแพร่หลายต่อไปในอนาคต

1.2 วัตถุประสงค์ของโครงงาน

- เพื่อพัฒนาระบบรักษาความปลอดภัยของการส่งข้อมูลผ่านเครือข่าย ให้มีความปลอดภัยและมีความซับซ้อนมากยิ่งขึ้น
- เป็นทางเลือกในการเพิ่มความปลอดภัยของข้อมูล โดยเน้นไปที่การเข้ารหัสลับข้อมูลก่อนส่งผ่านเครือข่าย ที่สามารถใช้งานได้ง่าย และสะดวกสบายอีกทางหนึ่ง

1.3 ขอบเขตของโครงงาน

- เครื่องคอมพิวเตอร์สองเครื่องสามารถส่งข้อมูลที่ผ่านการเข้ารหัสลับแล้วโต้ตอบกันได้
- ก่อนที่จะทำการเข้ารหัสลับหรือถอดรหัสลับข้อมูลจะต้องมีการเสียบบัตรเข้าไปที่เครื่องอ่านสมาร์ตการ์ด
- ภายในสมาร์ตการ์ดจะมีกุญแจรหัสที่ใช้ในการเข้ารหัสลับและถอดรหัสลับข้อมูลอยู่ ซึ่งทั้งทางผู้รับและผู้ส่งข้อมูลจะมีสมาร์ตการ์ดคนละใบ ที่ภายในมีกุญแจรหัสชุดเดียวกัน
- ภายในสมาร์ตการ์ดแต่ละใบ จะมีกุญแจรหัสจำนวนมาก และมีลำดับกำกับอยู่ ซึ่งการนำไปใช้แต่ละครั้งจะทำการสุ่มลำดับของกุญแจรหัสขึ้นมาใช้งาน
- มีส่วนของโปรแกรมที่สามารถทำหน้าที่เข้ารหัสลับและถอดรหัสลับข้อมูลที่อยู่ในรูปแบบของข้อความ (Clear Text) ได้

1.4 ขั้นตอนการดำเนินโครงงาน

1. ออกแบบวงจรของเครื่องอ่าน-เขียนสมาร์ตการ์ด
2. สร้างวงจรของเครื่องอ่าน-เขียนสมาร์ตการ์ด
3. ออกแบบโปรแกรมที่ใช้ควบคุมการทำงานของเครื่องอ่าน-เขียนสมาร์ตการ์ด
4. ออกแบบโปรแกรมติดต่อกับผู้ใช้ ซึ่งเป็นโปรแกรมที่ใช้ในการเข้ารหัสลับและถอดรหัสลับข้อมูลที่ต้องการจะส่ง
5. ทำการทดลอง บันทึกผลที่ได้ และวิเคราะห์ปัญหาที่เกิดขึ้น

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1. การป้องกันและรักษาความปลอดภัยของการส่งข้อมูลผ่านเครือข่ายมีความซับซ้อนมากยิ่งขึ้น ทำให้ยากต่อการลักลอบเข้าระบบหรือจากการโจมตีของผู้ประสงค์ร้าย

2. เกิดแนวความคิดและทางเลือกใหม่ในการรักษาความปลอดภัยของการส่งข้อมูลผ่านเครือข่าย

3. สามารถใช้ได้กับบุคคลใดก็ตาม ที่ต้องการส่งข้อมูลที่เป็นความลับ โดยไม่ต้องการให้ผู้อื่นที่ไม่ได้รับอนุญาตทำการอ่านข้อมูลได้ รวมไปถึงหน่วยงานหรือองค์กรต่างๆ ที่สามารถนำระบบนี้ไปใช้ได้อย่างมั่นใจ และมีประสิทธิภาพ

บทที่ 2
ทฤษฎี

2.1 สมาร์ตการ์ดคืออะไร

สมาร์ตการ์ด (Smart card) คือบัตรพลาสติกที่มีชิปไอซี (Integrated Circuit) ติดหรือฝังอยู่ในตัวบัตรพลาสติกตามมาตรฐาน ISO (International Standard Organization) เพื่อใช้ในการเก็บข้อมูลและประมวลผลภายในตัวเองโดยวิธีการเข้ารหัสลับตามมาตรฐาน DES Algorithm (Data Encryption Standard) เพื่อให้ระบบมีระดับความปลอดภัยสูงขึ้น ด้วยคุณสมบัติสำคัญประการหนึ่งที่ทำให้สมาร์ตการ์ดมีความแตกต่างจากบัตรพลาสติกทั่วไปก็คือ ขณะทำรายการ (Transaction) สมาร์ตการ์ดสามารถทำงานได้ด้วยตัวของมันเอง โดยไม่ต้องอาศัยติดต่อสื่อสารกับระบบหลัก (Font End) นั้นก็คือสมาร์ตการ์ดไม่จำเป็นต้องมีการติดต่อสื่อสารกับศูนย์กลางข้อมูลเหมือนกับบัตรแถบแม่เหล็ก (Off-line) ทำให้ประหยัดในเรื่องระบบสื่อสารไปได้มาก

2.2 ประวัติความเป็นมาของสมาร์ตการ์ด

สมาร์ตการ์ดปรากฏขึ้นครั้งแรกในประเทศเยอรมัน ในปี 1968 โดยชาวเยอรมัน (Jurgen Dethloff และ Helmut Grotupp) เป็นผู้คิดค้น แต่ผู้ที่ได้มาซึ่งสิทธิบัตรกลับเป็นชาวญี่ปุ่น (Kunitaka Arimura) ในปี 1970 และมีการจดสิทธิบัตรในชื่อของสมาร์ตการ์ดโดยชาวฝรั่งเศส (Roland Moreno) ในปี 1974 ในระยะแรกนั้นสมาร์ตการ์ดยังทำงานได้ไม่สมบูรณ์นัก เพราะสมาร์ตการ์ดรุ่นแรก ๆ ยังมีปัญหาทางเทคนิคเล็ก ๆ น้อย ๆ ซึ่งถึงแม้ว่าสมาร์ตการ์ดจะถือกำเนิดในยุโรป แต่ในระยะแรกสมาร์ตการ์ดกลับไม่ค่อยได้รับความสนใจเท่าที่ควร จนกระทั่งปี 1984 บริษัท French PTT (Postal and Telecommunications Services) ได้นำสมาร์ตการ์ดมาใช้งานเป็นบัตรโทรศัพท์ เพื่อป้องกันการโกงค่าโทรศัพท์ ซึ่งในครั้งนั้นถือว่าเป็นโครงการนำร่อง โดยมีการนำบัตรแถบแม่เหล็ก บัตรแถบแสง (Optical Storage) และสมาร์ตการ์ดมาทำการทดลองใช้งานเปรียบเทียบกัน ซึ่งแน่นอนว่าสมาร์ตการ์ดได้พิสูจน์ให้เห็นคุณลักษณะที่เหนือกว่าบัตรชนิดอื่น ทั้งในเรื่องของความทนทาน ความปลอดภัย ความสวยงาม เป็นผลให้สมาร์ตการ์ดในรูปของบัตรโทรศัพท์มีการนำไปใช้ถึง 60 ล้านใบ (เฉพาะประเทศฝรั่งเศส) และตอกย้ำความสำเร็จอีกกว่า 100 ล้านใบจาก 50 ประเทศทั่วโลกในปี 1997 กระนั้นสมาร์ตการ์ดก็ยังเป็นเพียงบัตรโทรศัพท์ การนำสมาร์ตการ์ดมาใช้ทางด้านการเงินการธนาคารกลับเป็นไปอย่างเชื่องช้า เนื่องจากบัตรที่เกี่ยวข้องกับระบบการเงินการธนาคารมีความยุ่งยากมากกว่าบัตรโทรศัพท์

และในปี 1960 เทคโนโลยีการประมวลผลเพื่อเข้ารหัสลับข้อมูลของฮาร์ดแวร์และซอฟต์แวร์มีความพร้อมมากขึ้น จึงมีการนำมาใช้ในการเข้ารหัสลับข้อมูลในสมาร์ตการ์ด ซึ่งแต่เดิมนั้นการเข้ารหัสลับจะมีการใช้งานเฉพาะในหน่วยงานทหาร หรือหน่วยงานราชการลับเท่านั้น ด้วยเหตุนี้ทำให้สมาร์ตการ์ดสามารถทำการเข้ารหัสลับ–ถอดรหัสลับข้อมูลได้ด้วยตัวมันเอง ทำให้การใช้สมาร์ตการ์ดมีความปลอดภัยสูงขึ้นจนสามารถนำมาใช้เป็นบัตรเครดิต หรือบัตรเงินสดได้อย่างสมบูรณ์แบบ

ในปี 1984 ธนาคารในฝรั่งเศสได้นำสมาร์ตการ์ดมาใช้เป็นบัตรเครดิตเป็นครั้งแรก ในระยะแรกนั้นต้องประสบกับปัญหามากมาย เกี่ยวกับการเข้ากันได้ของบัตรต่างธนาคาร ซึ่งต้องใช้เวลาถึง 10 ปีที่จะทำให้เข้ากันได้ทั้งหมด เป็นเหตุให้มีการรวมกันของ Europay, VISA และ MASTER เพื่อกำหนดมาตรฐานแก่เครดิตการ์ด ในรูปของสมาร์ตการ์ดให้มีมาตรฐานเดียวกันทุกธนาคารในชื่อของมาตรฐาน EMV (Europay, MASTER, VISA) โดยอ้างอิงกับมาตรฐาน ISO7816 เป็นหลัก ทำให้มีผู้ที่ต้องการพัฒนาแอพพลิเคชั่นเครดิตหรือเดบิตบนสมาร์ตการ์ด ต้องทำตามข้อกำหนดของมาตรฐาน EMV เท่านั้น

2.3 ส่วนประกอบและโครงสร้างของสมาร์ตการ์ด

2.3.1 ตัวบัตรพลาสติก

สมาร์ตการ์ดเป็นชิปไอซีขนาดเล็กที่ถูกสร้างขึ้นเช่นเดียวกับชิ้นส่วนอิเล็กทรอนิกส์อื่น ๆ ที่สร้างจากสารกึ่งตัวนำ นำมาติดลงบนหน้าสัมผัส และทำการฝังลงในเนื้อบัตรพลาสติก ซึ่งพลาสติกที่นิยมนำมาทำเป็นตัวบัตรจะใช้พลาสติก 4 ชนิด ได้แก่ PVC (Polyvinyl Choloride), ABS (Acrylonitrile Butadiene Styrene), PC (Polycarbonate) และ PET (Polyethylene Terephthalate) ในประเทศไทยจะใช้บัตรพลาสติก PVC มากเป็นอันดับหนึ่ง ส่วนอันดับสองเป็นบัตรพลาสติกชนิด ABS ซึ่งบัตรพลาสติกชนิด PVC มักนำมาใช้เป็นบัตรเอทีเอ็ม บัตรเครดิต–เดบิต บัตรประจำตัวประชาชน ฯลฯ ส่วนบัตรพลาสติกชนิด ABS ไม่ค่อยพบว่าใช้งานกันมากนักเนื่องจากราคาสูงกว่าและลายที่พิมพ์ลงบนบัตรไม่สวยงามคงทนเท่าบัตรพลาสติกชนิด PVC จะพบก็เพียงบัตรพลาสติกเนื้อผสมโดยใช้พลาสติกชนิด ABS เป็นแกนและฉาบผิวด้วยพลาสติกชนิด PVC แต่ความทนทานของตัวบัตรจะสู้บัตรพลาสติกเนื้อ PVC ล้วนไม่ได้

สำหรับบัตรพลาสติกอีก 2 ชนิดที่เหลือ ยังไม่พบว่ามีการใช้งานในประเทศไทย อาจเนื่องมาจากราคาที่สูงเกินไปของวัสดุที่นำมาใช้ทำเป็นตัวบัตร และคุณสมบัติของวัสดุที่ด้อยกว่าพลาสติกชนิด PVC แต่ข้อเสียที่สำคัญของพลาสติกชนิด PVC ก็ไม่ด้อยไปกว่าข้อดีของมัน นั้นก็คือมันไม่สามารถย่อยสลายได้ตามธรรมชาติ ซึ่งเท่ากับเป็นขยะสำหรับสิ่งแวดล้อมเลยทีเดียว

2.3.2 หน้าสัมผัสและชิปสมาร์ตการ์ด (Smart card Module)

สมาร์ตการ์ดโมดูล หรือ หน้าสัมผัสและชิปสมาร์ตการ์ด คือ ส่วนที่แสดงความเป็นตัวตนของสมาร์ตการ์ดที่ชัดเจนที่สุด สมาร์ตการ์ดบางชนิดเมื่อหยิบขึ้นมาเราอาจไม่ทราบได้เลยว่ามันคือสมาร์ตการ์ดที่มีการฝังชิปไว้ในเนื้อบัตร ดังนั้นการที่จะระบุกว่าบัตรใบใดเป็นบัตรสมาร์ตการ์ดนั้นต้องดูที่หลักการทำงานและลูกเล่นของบัตรเป็นหลัก ซึ่งต้องใช้ประสบการณ์ที่เกี่ยวกับสมาร์ตการ์ดพอสมควร แต่ในที่นี้จะขอแนะนำให้เห็นภาพลักษณ์ที่ชัดเจนของสมาร์ตการ์ดเป็นหลัก ซึ่งก็คือส่วนของสมาร์ตการ์ดโมดูลนั่นเอง

ในการผลิตสมาร์ตการ์ดโมดูล ส่วนที่เป็นหน้าสัมผัสของสมาร์ตการ์ดประกอบด้วยโลหะหลายชิ้นประกอบกัน แต่ละส่วนจะถูกยึดด้วยแถบฟิล์มบาง ๆ ทางด้านหลังของหน้าสัมผัสเพื่อให้คงรูปอยู่ได้ แถบฟิล์มตัวนี้จะมีการเจาะช่องเล็ก ๆ สำหรับการเชื่อมต่อสายนำสัญญาณกับชิปสมาร์ตการ์ดกับหน้าสัมผัส หลังจากที่วางชิปสมาร์ตการ์ดลงในตำแหน่งที่ต้องการและเชื่อมต่อสายนำสัญญาณจากชิปสมาร์ตการ์ดเข้ากับหน้าสัมผัสเรียบร้อยแล้ว ขั้นตอนสุดท้ายจะเป็นการผนึกชิปสมาร์ตการ์ดเพื่อป้องกันตัวชิป และสายนำสัญญาณต่าง ๆ จากสิ่งแวดล้อมภายนอก (เป็นการทดสอบขั้นต้น) ส่วนขั้นตอนที่เหลือจะเป็นการนำหน้าสัมผัสและชิปใส่ลงในบัตรพลาสติก และทดสอบการทำงานของชิปขั้นสุดท้าย

2.4 รายละเอียดพื้นฐานเกี่ยวกับสมาร์ตการ์ด

สมาร์ตการ์ดเป็นบัตรพลาสติกขนาดเท่าบัตรเครดิต หรือบัตรเอทีเอ็ม (ATM: Automatic Teller Machine) ที่มีหน่วยเก็บข้อมูล และหน่วยประมวลผลที่เรียกว่า ไมโครชิป ติดอยู่บนบัตร ซึ่งข้อมูลนี้อาจจะอยู่ในรูปของตัวเลขหรือตัวอักษรก็ได้ โดยมีกลไกในการเขียนและการอ่านข้อมูลที่ซับซ้อน ทำให้ยากต่อการปลอมแปลง จึงสามารถนำมาใช้ประโยชน์ในด้านต่าง ๆ มากมาย เช่น ด้านการเงินและการธนาคาร ด้านโทรคมนาคม ด้านงานทะเบียน ด้านการศึกษา ด้านการรักษาความปลอดภัย เป็นต้น

รูปที่ 2-1 บัตรสมาร์ตการ์ด

สมาร์ตการ์ดมีพื้นฐานมาจากระบบไมโครโปรเซสเซอร์ ซึ่งมีแนวคิดเริ่มแรกจากการนำชิปหน่วยความจำ (EEPROM) มาฝังลงในบัตรพลาสติก โดยมีหน้าสัมผัสเป็นขาเชื่อมต่อกับระบบภายนอก ในการเชื่อมต่อต้องมีการป้อนกระแสไฟฟ้าให้ชิปหน่วยความจำสามารถทำงานได้ การสั่งงานเพื่ออ่านหรือเขียนข้อมูลจากชิปหน่วยความจำสมาร์ตการ์ด ก็ทำได้โดยการเชื่อมต่อสัญญาณผ่านหน้าสัมผัสที่กำหนดไว้แล้ว ในการเชื่อมต่อขาสัญญาณของชิปหน่วยความจำแบบธรรมดา อาจไม่เหมาะสมนักสำหรับบัตรพลาสติกขนาดเล็ก เนื่องจากจำนวนขาสัญญาณของหน่วยความจำ (Bus) นั้นมีจำนวนมาก ยิ่งในหน่วยความจำที่มีความจุสูง ๆ ยิ่งต้องใช้สัญญาณอ้างอิงตำแหน่งของข้อมูล (Address Bus) มากขึ้น จึงมีการนำเอาระบบสื่อสารแบบซิงเกิลบัสมาใช้ในการรับส่งข้อมูล โดยในการนำเอาระบบสื่อสารแบบอนุกรมมาใช้ จำเป็นต้องมีการป้อนสัญญาณนาฬิกาเพื่อกำกับจังหวะการรับ-ส่งข้อมูลแต่ละบิต ทำให้ต้องมีหน้าสัมผัสสำหรับสัญญาณนาฬิกาบนชิปสมาร์ตการ์ดเพิ่มขึ้นมา แต่ก็นับว่าทำให้ขาเชื่อมต่อลดลงไปไม่น้อยทีเดียว ด้วยเหตุนี้สมาร์ตการ์ดชนิดหน่วยความจำจึงเป็นสมาร์ตการ์ดชนิดแรกที่ถูกสร้างขึ้น

การนำเอาชิปหน่วยความจำมาใส่ในบัตรพลาสติก ทำให้เกิดข้อดีเหนือบัตรแถบแม่เหล็กด้วยความจุข้อมูลที่มากกว่า ไม่มีผลต่อสนามแม่เหล็กไฟฟ้า และรอยขีดข่วน ทำให้สมาร์ตการ์ดโดดเด่นกว่าบัตรแถบแม่เหล็กอย่างเทียบกันไม่ได้ แต่ข้อเสียประการหนึ่งของการใช้หน่วยความจำเพียงอย่างเดียวคือ สามารถทำการอ่านและเขียนข้อมูลได้อย่างอิสระเช่นเดียวกับบัตรแถบแม่เหล็ก จึงถือได้ว่าความปลอดภัยของข้อมูลเกือบเป็นศูนย์ นั่นก็คือ ข้อมูลภายในสมาร์ตการ์ดชนิดนี้ไม่เป็นความลับ ด้วยเหตุนี้จึงมีการเพิ่มวงจรสำหรับป้องกันลงไปอีก เพื่อให้ผู้ออกบัตร (Card Issue) สามารถกำหนดสิทธิในการเข้าถึงข้อมูลแต่ละไบต์ด้วยวงจรฟิวส์แมทริกธรรมดา ๆ ที่เมื่อกำหนดเงื่อนไขไปแล้วจะไม่สามารถแก้ไขได้อีก ต่อมาเมื่อเทคโนโลยีทางด้านเซมิคอนดักเตอร์สูงขึ้น จึงมีการออกแบบวงจรที่สามารถกำหนดเป็นกุญแจรหัส (PIN) สำหรับเข้าถึงข้อมูลในบัตร ซึ่งต้องทำการแสดงกุญแจรหัสทุกครั้งที่บัตรเริ่มทำงาน เพื่อป้องกันการเจาะระบบอีกชั้นหนึ่ง อีกทั้งกุญแจรหัสก็ยังสามารถเปลี่ยนแปลงได้อีกด้วย

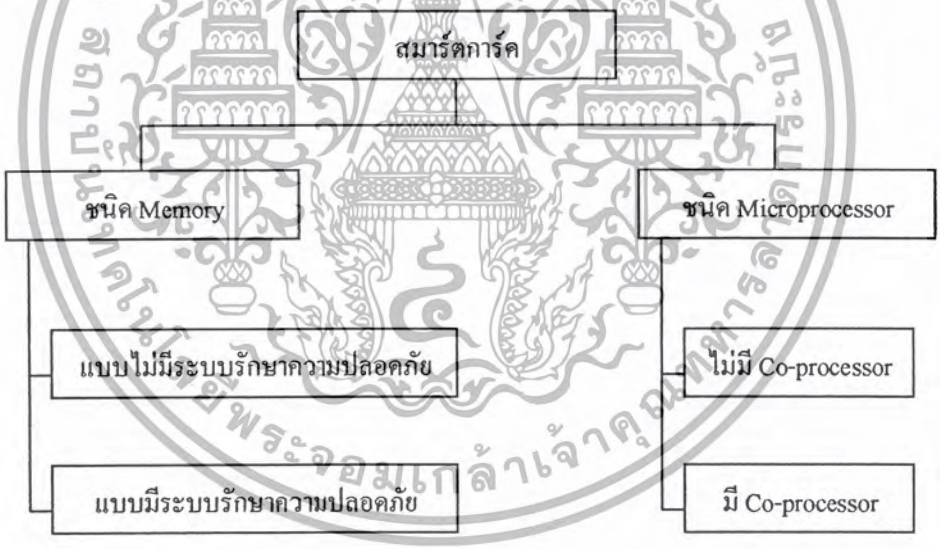
ต่อมามีการนำเอาไมโครโปรเซสเซอร์ (ที่จริงแล้วเป็นไมโครคอนโทรลเลอร์ แต่ในที่นี้จะขอเรียกว่าไมโครโปรเซสเซอร์เป็นหลัก) มาใส่ลงในสมาร์ตการ์ด ทำให้เกิดเป็นสมาร์ตการ์ดชนิดใหม่ที่มีความซับซ้อนยิ่งขึ้น การเข้าถึงข้อมูลไม่สามารถทำได้โดยตรงเหมือนอย่างสมาร์ตการ์ดชนิดหน่วยความจำ การใช้งานสมาร์ตการ์ดชนิดนี้ต้องเขียนขึ้นเป็นชุดคำสั่ง และส่งให้กับชิปไมโครโปรเซสเซอร์ทำงานแทน การที่ใส่ชิปไมโครโปรเซสเซอร์ลงไปในสมาร์ตการ์ด ทำให้ต้องมีการเพิ่มส่วนของหน่วยความจำโปรแกรม (OS-Operating System) สำหรับไมโครโปรเซสเซอร์

เพื่อให้ไมโครโปรเซสเซอร์สามารถทำการประมวลผลคำสั่งต่าง ๆ และสามารถโปรแกรมการเข้าถึงข้อมูล ทำให้ช่องโหว่ที่สำคัญของสมาร์ตการ์ดได้รับการแก้ไขจนเกือบสมบูรณ์แบบ

นอกจากสมาร์ตการ์ดทั้งสองชนิดที่ได้กล่าวมา ยังมีสมาร์ตการ์ดอีกชนิดหนึ่งที่ไม่ใช้หน้าสัมผัส (Contactless) ในการรับส่งสัญญาณ โดยอาศัยเทคโนโลยีคลื่นวิทยุในการติดต่อสื่อสาร สมาร์ตการ์ดชนิดนี้อาศัยการแปลงคลื่นวิทยุส่วนหนึ่งมาใช้เป็นกระแสไฟฟ้าสำหรับป้อนให้ชิป อีกส่วนหนึ่งมาดีเทกเอาข้อมูลคำสั่งให้ชิป สมาร์ตการ์ดชนิดนี้ได้รับความนิยมค่อนข้างมาก เพราะความน่าตื่นตาตื่นใจ และความล้ำสมัยของมัน แต่กระนั้นราคาของมันก็ย่อมสูงตามไปด้วย

2.5 ชนิดของสมาร์ตการ์ด

การแบ่งชนิดของสมาร์ตการ์ดในปัจจุบันอาจทำได้ยากสักหน่อย เนื่องจากมีการใส่เทคโนโลยีใหม่ ๆ ลงสมาร์ตการ์ดตลอดเวลา ถ้าจะแบ่งตามชนิดของหน่วยความจำภายในอาจไม่ชัดเจนนัก ยิ่งแบ่งตามลักษณะการเชื่อมต่อก็คงไม่ครอบคลุมสมาร์ตการ์ดทั้งหมด ดังนั้นจึงขอแสดงการแบ่งชนิดของสมาร์ตการ์ดให้เข้าใจได้ง่ายๆ ดังรูปที่ 2-2



รูปที่ 2-2 การแบ่งสมาร์ตการ์ดตามชนิดของหน่วยความจำ

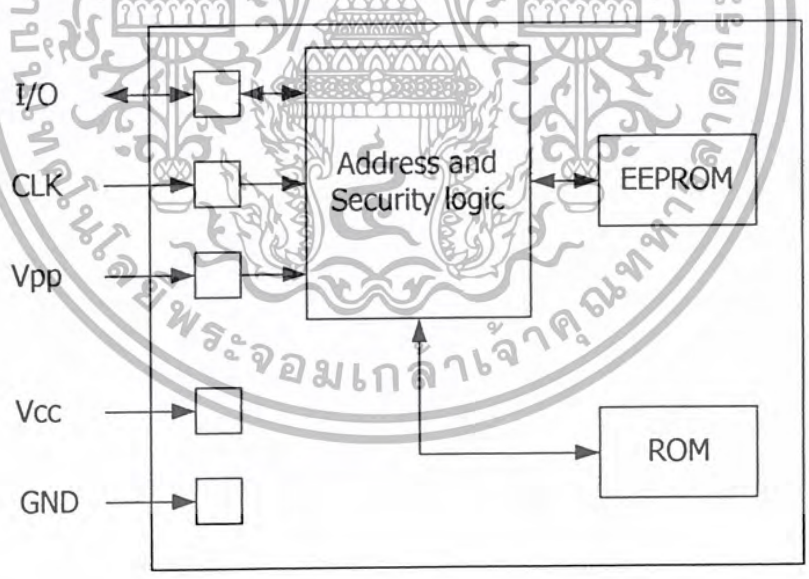
จากรูปที่ 2-2 จะเห็นได้ว่าเราสามารถแบ่งสมาร์ตการ์ดจากโครงสร้างภายในได้ 2 ชนิดก็คือ สมาร์ตการ์ดชนิดหน่วยความจำ (Memory Card) และ สมาร์ตการ์ดชนิดไมโครโปรเซสเซอร์ (Processor Card) ซึ่งชิปทั้งสองแบบจะมีหน้าสัมผัสเหมือนกัน แต่สัญญาณที่ต้องป้อนให้แก่หน้าสัมผัสบางหน้าสัมผัส จะไม่มีการใช้งานในสมาร์ตการ์ดต่างชนิดกัน เช่น แรงดันไฟฟ้าสำหรับ

การเขียนข้อมูลลงในชิป (Vpp) จะมีใช้ในสมาร์ตการ์ดชนิดหน่วยความจำเท่านั้น, สัญญาณนาฬิกาสำหรับป้อนให้ชิปทำงาน (CLK) ต้องป้อนให้กับชิปเหมือนกัน สำหรับสัญญาณนาฬิกา (CLK) ที่ป้อนให้ชิปสมาร์ตการ์ดเป็นสัญญาณนาฬิกาภายนอกที่ป้อนให้ชิปทำงานได้ เพราะภายในชิปสมาร์ตการ์ดไม่มีวงจรสำหรับสร้างสัญญาณนาฬิกา แต่หน้าสัมผัส I/O จะมีการรับ-ส่งข้อมูลที่แตกต่างกันในเรื่องของความถี่ และวิธีการควบคุมจังหวะการรับ-ส่งของข้อมูลแต่ละบิต

ในการแบ่งสมาร์ตการ์ดออกเป็น 2 ชนิด ตามชนิดของวงจรภายในดังที่กล่าวมา อาจแบ่งได้อีกลักษณะคือ แบ่งตามความถี่ในการรับ-ส่งข้อมูลผ่านหน้าสัมผัส I/O ของสมาร์ตการ์ด ดังที่กล่าวไปแล้ว ซึ่งสามารถแบ่งได้ดังนี้

2.5.1 การ์ดหน่วยความจำ (Memory card)

สมาร์ตการ์ดชนิดหน่วยความจำ (Memory) หรืออีกชื่อหนึ่งคือ Synchronous card เนื่องจากสมาร์ตการ์ดชนิดนี้มีการรับ-ส่งข้อมูลตามสัญญาณนาฬิกาที่ป้อนให้แก่ชิป (ข้อมูลแต่ละบิตที่ส่งให้แก่ชิปต้องสัมพันธ์กับสัญญาณนาฬิกา) สมาร์ตการ์ดชนิดนี้มีโครงสร้างที่ประกอบไปด้วย ส่วนวงจรสำหรับการติดต่อสื่อสารกับภายนอก หน่วยความจำข้อมูล และหน่วยความจำสำหรับเก็บชุดคำสั่งของสมาร์ตการ์ด ดังรูปที่ 2-3



รูปที่ 2-3 บล็อกไดอะแกรมโครงสร้างภายในชิปสมาร์ตการ์ดชนิดหน่วยความจำ

สมาร์ตการ์ดที่เป็นพื้นฐานของสมาร์ตการ์ดในปัจจุบัน ก็คือสมาร์ตการ์ดชนิด Free Access Memory สมาร์ตการ์ดชนิดนี้เปิดโอกาสให้อ่านหรือเขียนข้อมูลในแอดเดรสใด ๆ ก็ได้ตามชื่อของสมาร์ตการ์ดชนิดนี้ ไม่มีการป้องกันข้อมูลใด ๆ ภายในสมาร์ตการ์ดชนิดนี้ ซึ่งแน่นอนว่าเป็นสมาร์ตการ์ดที่มีความปลอดภัยต่ำที่สุด ถึงกระนั้นการอ่านข้อมูลก็ไม่ใช่เรื่องง่ายนักเมื่อมีการออกแบบหน่วยความจำข้อมูลให้มีการสลับตำแหน่งของบิตข้อมูล โดยมีวงจรควบคุมการสลับตำแหน่งของบิตเป็นส่วนป้องกันข้อมูลอีกต่อหนึ่ง ดังนั้นการอ่านข้อมูลออกแบบธรรมดาจะไม่ได้ข้อมูลที่ถูกต้องหากไม่ติดต่อกับวงจรควบคุมการสลับตำแหน่งของบิตโดยตรง

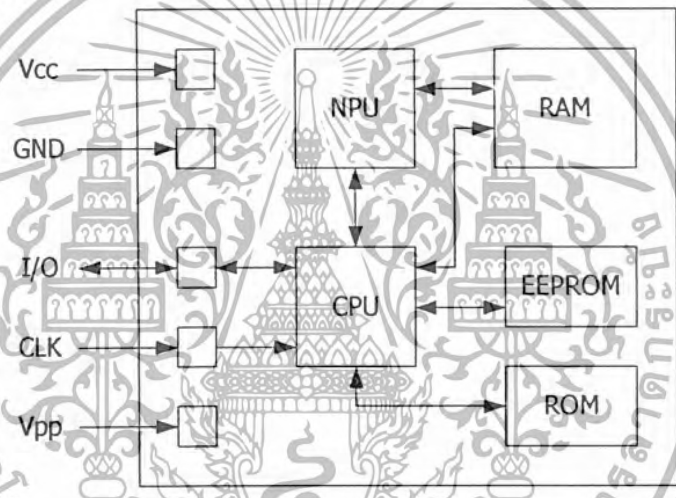
นอกจากนี้สมาร์ตการ์ดชนิดหน่วยความจำแบบธรรมดา ยังมีการใส่วงจรกำหนดเงื่อนไขการอ่านเขียนข้อมูลลงไปด้วย ทำให้สามารถกำหนดเงื่อนไขการอ่าน-เขียนข้อมูลได้ทุกไบต์ โดยสมาร์ตการ์ดที่มีวงจรป้องกันการอ่าน-เขียนชนิดนี้ถูกเรียกว่า PIN Protect Memory เนื่องจากการเข้าถึงข้อมูลจะต้องแสดงรหัสผ่านให้บัตรทราบก่อนจึงจะสามารถเข้าถึงข้อมูลได้ วงจรกำหนดเงื่อนไขการอ่าน-เขียนข้อมูลจะมีบิตพิเศษที่มีชื่อว่า Bit Protect ซึ่งเป็นบิตข้อมูลที่ฝากไว้กับข้อมูลให้เป็นบิตที่ 9 แต่ไม่สามารถแก้ไขได้ด้วยคำสั่งเขียนข้อมูลธรรมดา เพราะ Bit Protect ไม่ได้เป็นส่วนหนึ่งของข้อมูลจริง ๆ ในการแก้ไข Bit Protect นี้จะสามารถทำการเปลี่ยนแปลงได้เพียงครั้งเดียวด้วยคำสั่งเฉพาะเท่านั้น เช่น หากต้องการบังคับไม่ให้ข้อมูลไบต์ใดไม่สามารถแก้ไขได้ก็ให้ทำการเคลียร์บิตที่ 9 ของข้อมูลไบต์นั้น ๆ แต่สำหรับรหัสผ่านในการเข้าถึงข้อมูลสามารถเปลี่ยนแปลงได้ แต่ต้องแสดงรหัสผ่านชุดเก่าให้บัตรได้ทราบเสียก่อนจึงจะสามารถเปลี่ยนแปลงรหัสผ่านได้

สมาร์ตการ์ดอีกชนิดหนึ่งที่มีใช้เป็นบัตรโทรศัพท์ในประเทศไทยนั้นคือ การ์ดหน่วยความจำชนิด Token ภายในสมาร์ตการ์ดชนิดนี้ จะมีการเก็บข้อมูลในลักษณะจำนวนนับ (Counter) ซึ่งจำนวนนับนี้จะเป็นตัวเลขแทนมูลค่าของเงินที่ระบุบนบัตร การนับเลขเป็นการนับถอยหลังเพื่อเป็นการนับมูลค่าที่คงเหลือในบัตร หมายความว่าหากใช้บัตรในการโทรศัพท์ไปเรื่อย ๆ มูลค่าในบัตรก็จะถูกลดลงตามไปด้วยเช่นกัน ในการเข้าถึงข้อมูลของสมาร์ตการ์ดชนิดนี้ต้องมีการแสดงรหัสผ่านให้บัตรทราบเหมือนกับการ์ดหน่วยความจำชนิด PIN Protect แต่ไม่มี Bit Protect เท่านั้นเอง

สมาร์ตการ์ดชนิดหน่วยความจำเป็นสมาร์ตการ์ดพื้นฐานของสมาร์ตการ์ดรุ่นใหม่ ๆ ในปัจจุบัน ด้วยโครงสร้างและการทำงานที่ง่ายต่อการทำความเข้าใจ ราคาถูก สามารถเก็บข้อมูลได้จำนวนมาก และความเร็วในการทำงานของชิปไม่สูงนัก จึงทำให้สมาร์ตการ์ดชนิดนี้เหมาะที่จะนำไปประยุกต์ใช้กับงานที่ข้อมูลไม่ค่อยสำคัญมากนัก เช่น บัตรลงเวลาทำงาน บัตรผ่านประตู บัตรโทรศัพท์ ฯลฯ ปัจจุบันสมาร์ตการ์ดชนิดหน่วยความจำมีขนาดหน่วยความจำสูงสุดถึง 64 กิโลไบต์ และอีกไม่นานนักเราจะได้เห็นสมาร์ตการ์ดที่มีขนาดหน่วยความจำข้อมูลถึง 128 กิโลไบต์

2.5.2 การ์ดชนิดโปรเซสเซอร์ (Processor card)

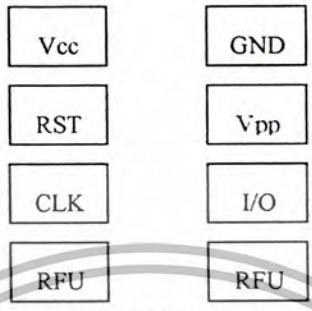
สมาร์ตการ์ดชนิดโปรเซสเซอร์ หรือเรียกอีกชื่อหนึ่งว่า Asynchronous card เป็นสมาร์ตการ์ดที่ได้รับการปรับปรุงจากสมาร์ตการ์ดชนิดหน่วยความจำ ด้วยการใส่เทคโนโลยีไมโครโปรเซสเซอร์เข้าไปในชิป เพื่อให้ชิปสามารถประมวลผลข้อมูล และเพิ่มความปลอดภัยให้แก่ข้อมูลได้สูงขึ้น จากการที่ใส่ไมโครโปรเซสเซอร์ลงไปในชิปนั้น ทำให้จำเป็นต้องมีการเพิ่มส่วนของหน่วยความจำไว้สำหรับจัดเก็บระบบปฏิบัติการของไมโครโปรเซสเซอร์ และหน่วยความจำชั่วคราวสำหรับการประมวลผลข้อมูล นอกจากนี้ยังมีการใส่ชิปประมวลผลทางคณิตศาสตร์ลงในชิปสมาร์ตการ์ด เพื่อช่วยให้การประมวลผลข้อมูลด้วยอัลกอริทึมสำหรับเข้ารหัสลับและถอดรหัสลับ ทำให้สมาร์ตการ์ดชนิดโปรเซสเซอร์มีความเร็วในการทำงานที่สูงกว่าสมาร์ตการ์ดชนิดหน่วยความจำหลายเท่า



รูปที่ 2-4 บล็อกไดอะแกรมโครงสร้างภายในชิปสมาร์ตการ์ดชนิดโปรเซสเซอร์

ในการรับส่งข้อมูลให้กับสมาร์ตการ์ดชนิดนี้ จะใช้หน้าสัมผัสเดียวกับสมาร์ตการ์ดชนิดหน่วยความจำ โดยสัญญาณนาฬิการที่ป้อนจะถูกใช้เป็นสัญญาณนาฬิกาให้แก่โปรเซสเซอร์ภายในสมาร์ตการ์ด ข้อมูลที่รับส่งจึงไม่จำเป็นต้องสัมพันธ์กับสัญญาณนาฬิกาที่ป้อนให้แก่ชิป เพียงกำหนดอัตราการรับ-ส่งข้อมูลเป็น 9,600 บิตต่อวินาที ก็สามารถติดต่อกับโปรเซสเซอร์ของชิปได้ แต่การเข้าถึงข้อมูลจะไม่สามารถทำได้เหมือนสมาร์ตการ์ดชนิดหน่วยความจำ การเข้าถึงข้อมูลต้องกระทำผ่านโปรเซสเซอร์ของสมาร์ตการ์ดเท่านั้น ไม่ว่าจะเป็นการอ่านหรือเขียนข้อมูลก็ตาม เพราะหน่วยความจำจะอยู่ภายในความควบคุมของโปรเซสเซอร์เพียงอย่างเดียว ข้อดีอย่างหนึ่งที่ไม่

สามารถติดต่อกับหน่วยความจำในชิปได้โดยตรงก็คือ การลอบเข้าถึงข้อมูลโดยไม่ได้รับอนุญาตแทบเป็นไปไม่ได้ ยกเว้นมีความบกพร่องในการกำหนดเงื่อนไขในการเข้าถึงข้อมูลที่เป็นความลับ



รูปที่ 2-5 ขาต่างๆ บนหน้าสัมผัสของบัตรสมาร์ตการ์ด

ชื่อขา	การใช้งาน
Vcc	แหล่งจ่ายกระแสไฟฟ้า
Vpp	สำหรับเขียนข้อมูลลงในชิป
GND	กราวนด์
RST	สัญญาณรีเซต
I/O	Input – Output สำหรับรับส่งข้อมูล
CLK	สัญญาณนาฬิกา

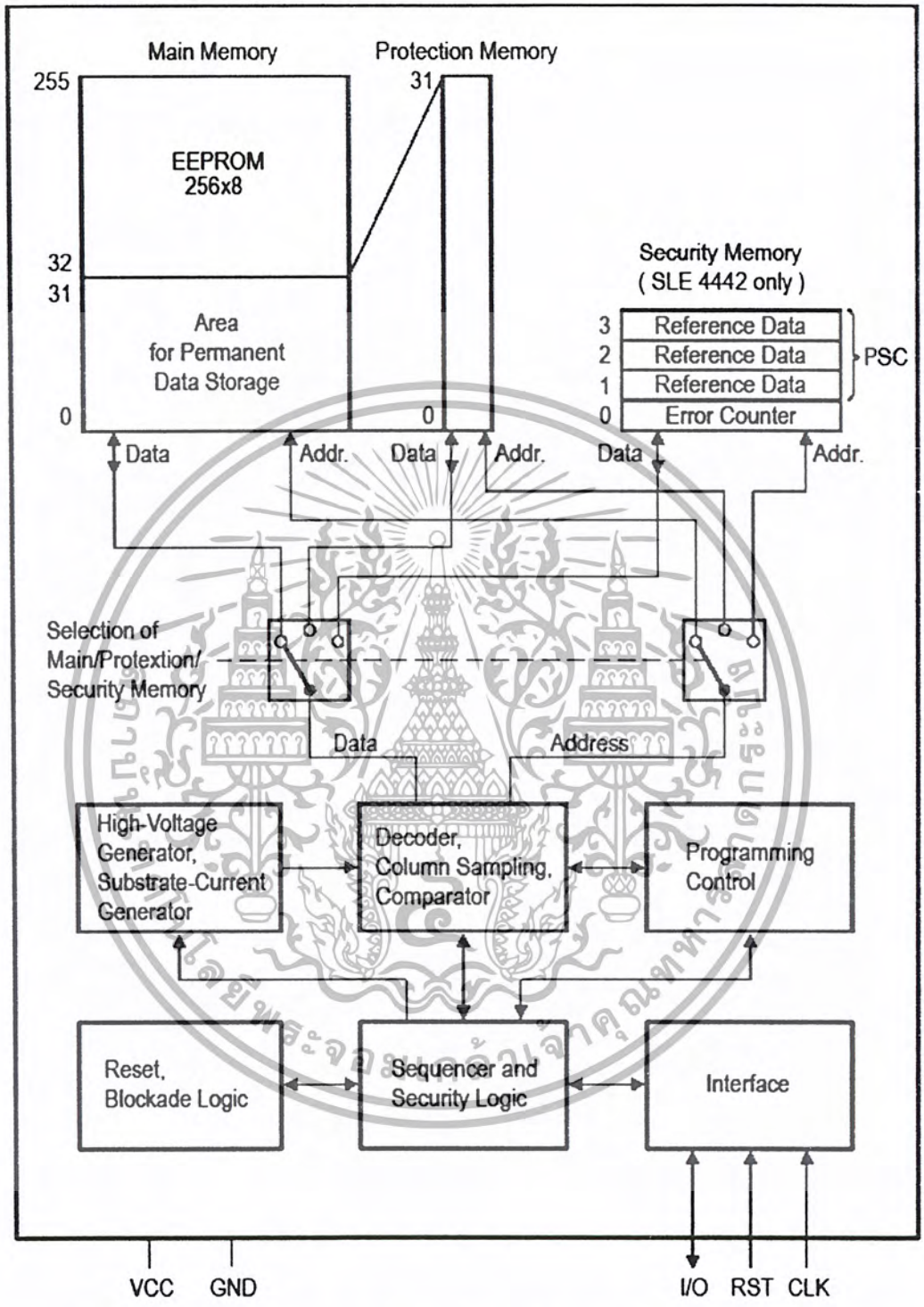
ตารางที่ 2.1 หน้าที่การทำงานของขาต่างๆ ของบัตรสมาร์ตการ์ด

2.6 การ์ดที่มีระบบป้องกันข้อมูล

การ์ดที่มีระบบป้องกันความปลอดภัยข้อมูล คือ สมาร์ตการ์ดที่การอ่านข้อมูลสามารถทำได้อย่างอิสระ แต่การเขียนข้อมูลจะไม่สามารถทำได้หากไม่มีรหัสผ่านที่ถูกต้อง วิธีการในลักษณะนี้ช่วยให้ข้อมูลภายในสมาร์ตการ์ดได้รับการปกป้องและมีความน่าเชื่อถือ รูปแบบการสื่อสารข้อมูลของสมาร์ตการ์ดชนิดนี้เป็นการสื่อสารข้อมูลแบบซิงโครนัส (Synchronous) ตามมาตรฐาน ISO7816 ซึ่งรูปแบบคำสั่งจะแตกต่างกันไปในผู้ผลิตแต่ละราย โดยในโครงงานนี้ได้เลือกใช้สมาร์ตการ์ดเบอร์ SLE4442 เนื่องจากเป็นการ์ดที่มีคุณสมบัติของการรักษาความปลอดภัยข้อมูลอย่างครบถ้วน และยังสามารถหามาใช้งานได้ง่ายในบ้านเรา

2.6.1 คุณสมบัติโดยทั่วไปของสมาร์ตการ์ดเบอร์ SLE4442

Smart Card เบอร์ SLE4442 มีหน่วยความจำแบบ EEPROM ขนาด 256 ไบต์ โดยแบ่งเป็น Protectable Data Memory 32 ไบต์ และ Unprotected Data Memory 224 ไบต์ สามารถอ่านและเขียนได้ 100,000 ครั้ง เก็บข้อมูลได้นานถึง 10 ปีส่วนที่เป็น Protectable Data Memory นั้นสามารถเขียนข้อมูลถาวรไว้โดยจะลบหรือแก้ไขเปลี่ยนแปลงไม่ได้อีกเลย และในส่วนนี้ได้ถูกเขียนข้อมูลถาวรไว้แล้ว 12 ไบต์ ตามมาตรฐาน ISO7816 นอกจากนี้ SLE4442 ยังมีPSC (Programmable Security Code) 3 ไบต์ เพื่อใช้ในการตรวจสอบค่าให้ตรงกับค่า PSC ที่มีในบัตรก่อนจึงจะเขียนข้อมูลลงในบัตรได้ และ EC (Error Counter) เพื่อใช้ในการนับจำนวนครั้งที่ทำการตรวจสอบ Verify ค่า PSC โดยถ้าทำการตรวจสอบ Verify ค่า PSC ไม่ถูกต้องถึง 3 ครั้ง บัตรนี้จะเขียนข้อมูลไม่ได้อีกเลยทันที การนับ Error Counter นี้จะถูก Reset เมื่อได้ทำการ Verify ค่า PSC ได้ถูกต้อง ค่า PSC มาตรฐานของบัตรใหม่ที่ผลิตจากโรงงานคือ FFFFFF

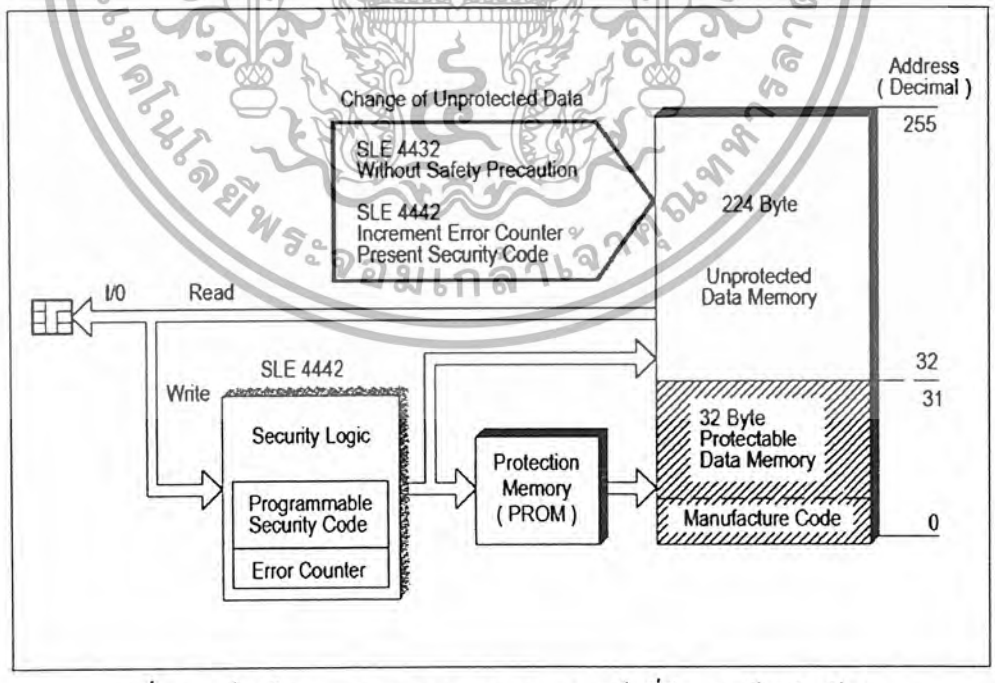


รูปที่ 2-6 บล็อกไดอะแกรมแสดงโครงสร้างภายในของสมาร์ตการ์ดเบอร์ SLE4442

จากรูปที่ 2-6 จะเห็นได้ว่าหน่วยความจำขนาด 256 ไบต์ ที่อยู่ภายในสมาร์ตการ์ดเบอร์ SLE4442 จะถูกแบ่งออกเป็น 2 ส่วนด้วยกัน ได้แก่ ข้อมูลในช่วง 32 ไบต์แรกซึ่งเป็นพื้นที่ที่มีระบบป้องกันการเขียนข้อมูลทับ และหน่วยความจำส่วนถัดมาซึ่งเป็นอีอีพรอม (EEPROM) ที่สามารถทั้งเขียนและอ่านได้ กลไกในการปกป้องข้อมูลของสมาร์ตการ์ดเบอร์ SLE4442 มาจากส่วนที่เป็นหน่วยความจำปลอดภัย (Security Memory) ที่ได้รับการปกป้องโดยข้อมูลสำคัญ 2 ส่วน คือ

- ข้อมูลอ้างอิง (Reference Data หรือ PSC) เป็นข้อมูลขนาด 3 ไบต์ ที่เก็บค่าของรหัสผ่านสำหรับการเข้าไปแก้ไขข้อมูลในหน่วยความจำเอาไว้ (รหัส PSC ไม่สามารถถูกอ่านออกมาได้) รหัส PSC จะถูกกำหนดเป็นค่าหนึ่งมาโดยผู้ผลิตก่อนซึ่งสามารถจะมาปรับเปลี่ยนเองได้ในภายหลังเมื่อใช้งาน

- ไบต์แสดงความผิดพลาด (Error Counter Byte) เป็นข้อมูลที่บอกถึงจำนวนครั้งที่ป้อนรหัส PSC ผิด ซึ่งถูกกำหนดเอาไว้ตายตัวว่าจะผิดได้ไม่เกิน 3 ครั้ง หากเกินกว่านั้นการ์ดจะล็อคตัวเองอย่างถาวรทันที และไม่มีทางปลดล็อคได้ แม้ว่าจะป้อนรหัส PSC ที่ถูกต้องไปแล้วก็ตาม การเขียนข้อมูลยังหน่วยความจำก็จะไม่สามารถทำได้อีกต่อไป แต่ยังคงอ่านข้อมูลออกมาได้ตามปกติ การป้อนรหัส PSC ผิดแต่ละครั้ง Error Counter จะถูกลดลง 1 ค่าทันที ถ้าหากค่า Error Counter ถูกลดลงจนมีค่าเป็น 0 เมื่อไรก็แสดงว่าการ์ดได้ถูกล็อคไปเรียบร้อยแล้ว (ในกรณีที่ป้อนรหัสถูกในครั้งที่ 3 ค่าของ Error Counter จะถูกรีเซตกลับไปเป็น 3 ครั้งเหมือนอย่างตอนแรกเริ่ม



รูปที่ 2-7 บล็อกไดอะแกรมแสดงภาพรวมของการ์ดที่มีระบบป้องกันข้อมูล

จากรูปที่ 2-7 จะเห็นได้ว่าการอ่านข้อมูลจากหน่วยความจำนั้น เราสามารถจะอ่านข้อมูลออกมาได้โดยไม่ต้องผ่านขั้นตอนของการป้อนรหัส PSC แต่สำหรับการเขียนข้อมูลแล้วเราจะต้องป้อนรหัส PSC ที่ถูกต้องเสียก่อน เพื่อเปิดลอจิกในการเขียนข้อมูลลงยังหน่วยความจำ นอกจากนั้นก็จะเห็นได้ว่าข้อมูล 4 ไบต์แรก เป็นข้อมูลของผู้ผลิตหรือ Manufacturer Code โดยพื้นที่ส่วนนี้ใช้เก็บข้อมูลของ ATR โดยความหมายของข้อมูลที่อยู่ในพื้นที่ส่วนนี้แต่ละไบต์จะถูกกำหนดโดยผู้ผลิตการ์ดแต่ละราย

2.6.2 รูปแบบการสื่อสารข้อมูลของสมาร์ตการ์ดเบอร์ SLE4442

รูปแบบการสื่อสารข้อมูลของสมาร์ตการ์ดเบอร์ SLE4442 เป็นการรับส่งข้อมูลระหว่างเครื่องอ่านและสมาร์ตการ์ดแบบ 2 ทิศทาง (ข้อมูลบนสาย I/O จะถูกอ่านค่าที่ขอบขาลงของสัญญาณนาฬิกา) โดยรูปแบบการสื่อสารนี้ประกอบด้วย 4 โหมดการทำงาน ได้แก่

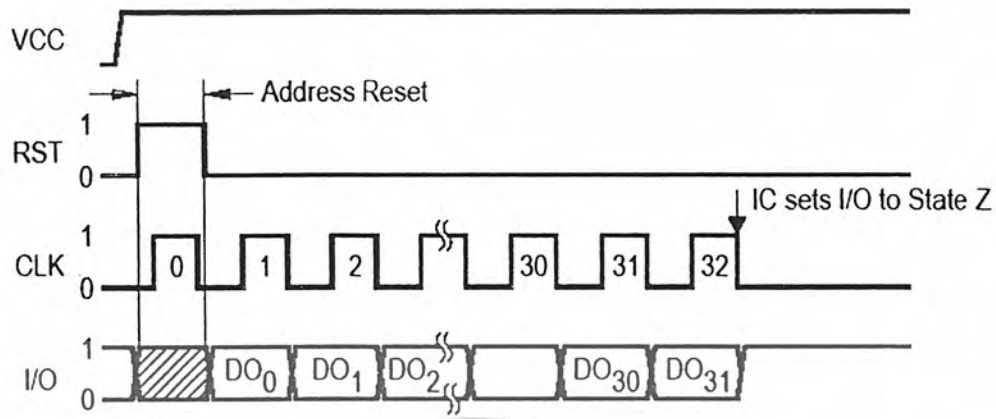
- การรีเซตและการตอบรับการรีเซตด้วย ATR (Answer To Reset)
- โหมดการส่งคำสั่ง (Command Mode)
- โหมดการอ่านข้อมูล (Outgoing Data Mode)
- โหมดการดำเนินการ (Processing Mode)

2.6.2.1 การรีเซตและการตอบรับการรีเซตด้วย ATR (Answer To Reset)

เมื่อรีเซตการทำงานของการ์ดจะทำให้การ์ดมีการตอบรับการรีเซตด้วยข้อมูล ATR สำหรับข้อมูลATR ที่ตอบกลับมาจากสมาร์ตการ์ดเบอร์ SLE4442 จะประกอบด้วยข้อมูล 4 ไบต์ การอ่านข้อมูลที่ว่านี้สามารถทำได้โดยอ้างอิงจากสัญญาณในรูปที่ 2-8 โดยหลังจากที่ขา RST เป็นลอจิกต่ำ เมื่อมีสัญญาณนาฬิกาลูกต่อไปเข้ามา จะทำให้เกิดสัญญาณเอาท์พุตของสมาร์ตการ์ดขึ้นที่ขา I/O ซึ่งก็คือ สัญญาณตอบรับการรีเซตนั่นเอง หลังจากที่ครบ 4 ไบต์แล้ว ที่ขา I/O จะเปลี่ยนเป็นลอจิกสูง เพื่อเป็นการบอกถึงการสิ้นสุดการรีเซต

Answer-to-Reset (Hex)	Byte 1	Byte 2	Byte 3	Byte 4
	$DO_7 \dots DO_0$	$DO_{15} \dots DO_8$	$DO_{23} \dots DO_{16}$	$DO_{31} \dots DO_{24}$

ตารางที่ 2.2 ลักษณะของข้อมูลที่ได้จากการตอบรับการรีเซต



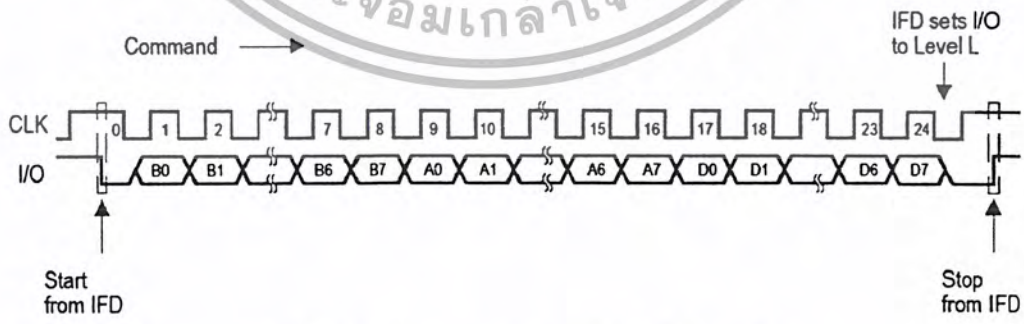
รูปที่ 2-8 รูปสัญญาณของการรีเซตและการตอบรับการรีเซตด้วย ATR

2.6.2.2 โหมดการส่งคำสั่ง (Command Mode)

การส่งคำสั่งไปยังสมาร์ตการ์ดหรือการทำงานในโหมดการส่งคำสั่ง (Command Mode) ก็คือ กระบวนการต่อเนื่องหลังจากการ์ดถูกรีเซตไปเรียบร้อยแล้ว โดยการ์ดจะรอรับคำสั่งที่ส่งมาจากเครื่องอ่านซึ่งมีรูปแบบเป็นข้อมูลมีความยาว 3 ไบต์ โครงสร้างของข้อมูลดังกล่าวประกอบด้วยคำสั่ง (Command), แอดเดรส (Address) และ ข้อมูล (Data) โดยคำสั่งทั้งหมดที่สมาร์ตการ์ดเบอร์ SLE4442 รองรับถูกแสดงดังตารางที่ 2.3 ส่วนรูปสัญญาณที่เกิดขึ้นระหว่างการทำงานของโหมดการส่งคำสั่งก็เป็นดังรูปที่ 2-9 จะเห็นได้ว่าการส่งข้อมูลแต่ละครั้งจะต้องมีการส่งสภาวะเริ่มต้นและสภาวะสิ้นสุดกำกับไปกับตัวข้อมูลด้วย โดยสภาวะเริ่มต้นก็คือการเปลี่ยนระดับจากลอจิกค่าสูงเป็นค่าต่ำที่ขา I/O ในขณะที่ระดับลอจิกที่ขา CLK เป็นค่าสูง ส่วนสภาวะสิ้นสุดก็คือการเปลี่ยนระดับจากลอจิกค่าต่ำเป็นสูงที่ขา I/O ในขณะที่ขา CLK เป็นค่าสูง

Byte 1 Control								Byte 2 Address	Byte 3 Data	Operation	Mode
B7	B6	B5	B4	B3	B2	B1	B0	A7-A0	D7-D0		
0	0	1	1	0	0	0	0	address	no effect	READ MAIN MEMORY	outgoing data
0	0	1	1	1	0	0	0	address	input data	UPDATE MAIN MEMORY	processing
0	0	1	1	0	1	0	0	no effect	no effect	READ PROTECTION MEMORY	outgoing data
0	0	1	1	1	1	0	0	address	input data	WRITE PROTECTION MEMORY	processing
0	0	1	1	0	0	0	1	no effect	no effect	READ SECURITY MEMORY	outgoing data
0	0	1	1	1	0	0	1	address	input data	UPDATE SECURITY MEMORY	processing
0	0	1	1	0	0	1	1	address	input data	COMPARE VERIFICATION DATA	processing

ตารางที่ 2.3 โครงสร้างและความหมายของชุดคำสั่งที่สมาร์ตการ์ดเบอร์ SLE4442 รองรับ



รูปที่ 2-9 รูปสัญญาณของการส่งคำสั่งไปยังการ์ด

MSB			Control				LSB	MSB			Address				LSB	MSB			Data				LSB
B7	B6	B5	B4	B3	B2	B1	B0	A7	A6	A5	A4	A3	A2	A1	A0	D7	D6	D5	D4	D3	D2	D1	D0

ตารางที่ 2.4 รูปแบบและส่วนประกอบของคำสั่ง

2.6.2.2.1 การอ่านข้อมูลจากหน่วยความจำหลัก (Read Main Memory)

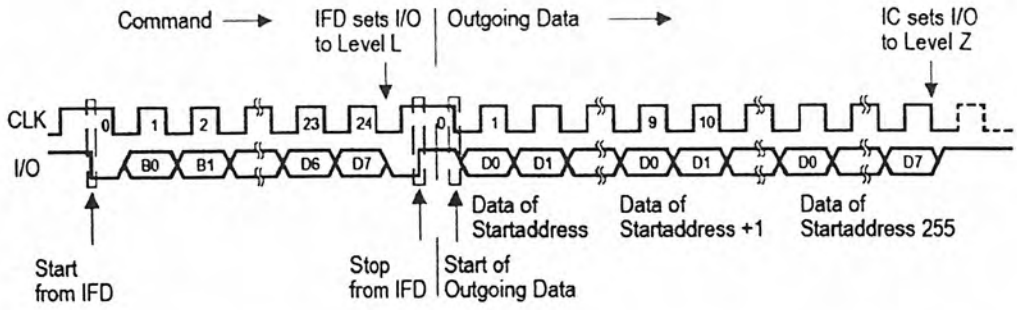
คือ คำสั่งที่ใช้ในการอ่านข้อมูลทั้งหมดออกมาจากหน่วยความจำของการ์ด ทั้งจากพื้นที่ส่วนที่ได้รับการป้องกัน (หน่วยความจำ 32 ไบต์แรก) และส่วนที่ไม่ได้รับการป้องกัน (หน่วยความจำ 224 ไบต์หลัง) โดยจะเป็นการอ่านค่าโดยเริ่มต้นจากแอดเดรสที่ส่งไปจนถึงแอดเดรสสุดท้าย (0FFH) ของพื้นที่หน่วยความจำ

Address (decimal)	Main Memory	Protection Memory	Security Memory (only SLE 4442)
255	Data Byte 255 (D7 ... D0)	–	–
:	:	–	–
32	Data Byte 32 (D7 ... D0)	–	–
31	Data Byte 31 (D7 ... D0)	Protection Bit 31 (D31)	–
:	:	:	–
3	Data Byte 3 (D7 ... D0)	Protection Bit 3 (D3)	Reference Data Byte 3 (D7 ... D0)
2	Data Byte 2 (D7 ... D0)	Protection Bit 2 (D2)	Reference Data Byte 2 (D7 ... D0)
1	Data Byte 1 (D7 ... D0)	Protection Bit 1 (D1)	Reference Data Byte 1 (D7 ... D0)
0	Data Byte 0 (D7 ... D0)	Protection Bit 0 (D0)	Error Counter

Command: READ MAIN MEMORY

	Control								Address	Data
	B7	B6	B5	B4	B3	B2	B1	B0	A7...A0	D7...D0
Binary	0	0	1	1	0	0	0	0	Address	No effect
Hexadecimal	30_H								$00_H...FF_H$	No effect

ตารางที่ 2.5 ลักษณะหน่วยความจำ และรูปแบบคำสั่งในการอ่านข้อมูลจากหน่วยความจำหลัก



รูปที่ 2-10 รูปสัญญาณของการอ่านข้อมูลจากหน่วยความจำหลัก

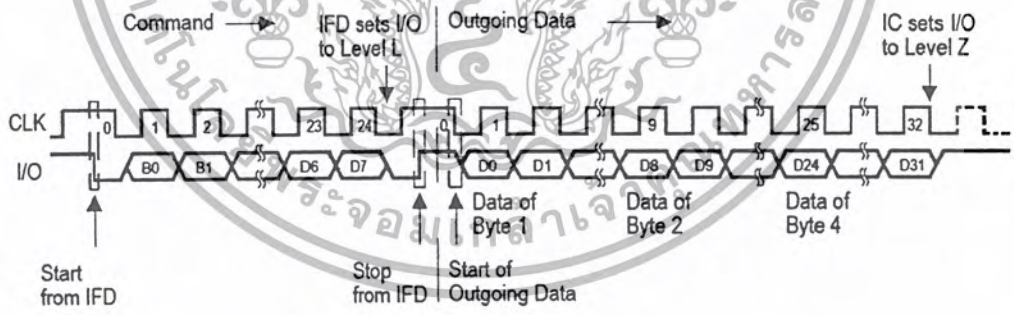
2.6.2.2.2 การอ่านข้อมูลจากหน่วยความจำที่มีการป้องกัน (Read Protection Memory)

คือ คำสั่งที่ใช้ในการอ่านข้อมูลทั้งหมดออกมาจากหน่วยความจำ 32 ไบต์แรก

Command: READ PROTECTION MEMORY

	Control								Address	Data
	B7	B6	B5	B4	B3	B2	B1	B0	A7…A0	D7…D0
Binary	0	0	1	1	0	1	0	0	No effect	No effect
Hexadecimal	34_H								No effect	No effect

ตารางที่ 2.6 รูปแบบคำสั่งในการอ่านข้อมูลจากหน่วยความจำที่มีการป้องกัน



รูปที่ 2-11 รูปสัญญาณของการอ่านข้อมูลจากหน่วยความจำที่มีการป้องกัน

2.6.2.2.3 การเขียนข้อมูลลงในหน่วยความจำหลัก (Update Main Memory)

คือคำสั่งที่ใช้ในการเขียนข้อมูลยังแอดเดรสใด ๆ ของหน่วยความจำทั้ง 256 ไบต์ ในกรณีที่ใช้คำสั่งนี้เขียนข้อมูลลงยังหน่วยความจำ 32 ไบต์แรก ข้อมูลจะยังคงแก้ไขเปลี่ยนแปลงได้ภายหลัง

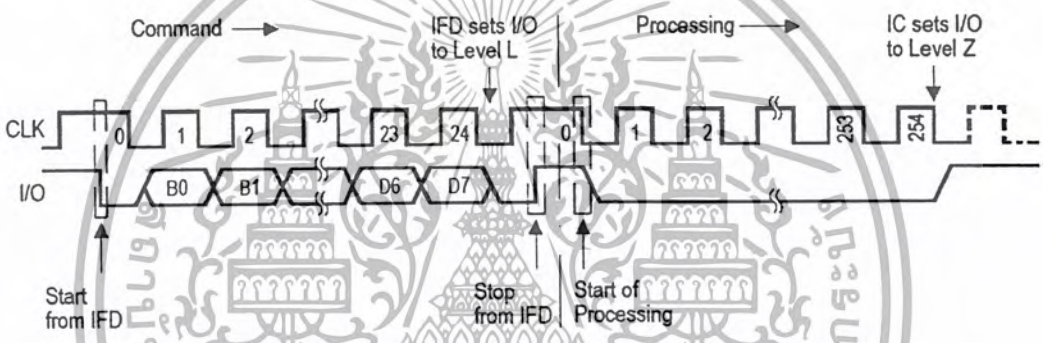
Command: UPDATE MAIN MEMORY

	Control								Address	Data
	B7	B6	B5	B4	B3	B2	B1	B0	A7...A0	D7...D0
Binary	0	0	1	1	1	0	0	0	Address	Input data
Hexadecimal	38_H								$00_H...FF_H$	Input data

ตารางที่ 2.7 รูปแบบคำสั่งในการเขียนข้อมูลลงในหน่วยความจำหลัก

สำหรับการเขียนข้อมูลจะประกอบด้วย 3 เงื่อนไข คือ

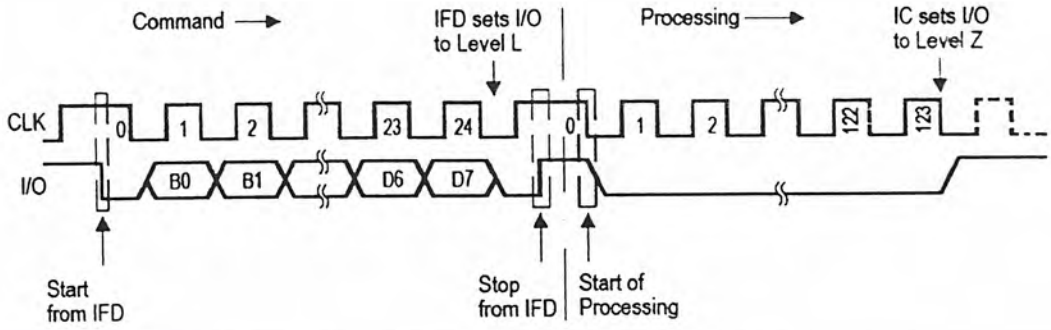
- การลบข้อมูลที่แอดเดรสของหน่วยความจำที่กำหนด ให้เป็น 0FFH แล้วทำการเขียนข้อมูลซ้ำลงยังแอดเดรสเดิม กระบวนการนี้ต้องใช้เวลา 5 มิลลิวินาที หรือเท่ากับสัญญาณนาฬิกา 255 ลูก



รูปที่ 2-12 รูปสัญญาณของการเขียนข้อมูลลงในหน่วยความจำหลัก แบบการลบข้อมูลแล้วเขียนข้อมูลซ้ำ

- การเขียนข้อมูลที่แอดเดรสของหน่วยความจำที่กำหนดโดยไม่ต้องลบข้อมูลออก สำหรับกรณีนี้แอดเดรสดังกล่าวจะต้องเป็นที่ว่าง (มีค่าข้อมูลเป็น 0FFH) อยู่ก่อนหน้านี้แล้วเท่านั้น กระบวนการนี้จะใช้เวลา 2.5 มิลลิวินาที หรือเท่ากับสัญญาณนาฬิกา 124 ลูก

- การลบข้อมูลที่แอดเดรสของหน่วยความจำที่กำหนด (ให้มีค่าข้อมูลเป็น 0FFH) โดยไม่มีการเขียนข้อมูลต่อ สำหรับกระบวนการนี้ใช้เวลา 2.5 มิลลิวินาที หรือเท่ากับสัญญาณนาฬิกา 124 ลูก



รูปที่ 2-13 รูปสัญญาณของการเขียนข้อมูลลงในหน่วยความจำหลัก
แบบการลบหรือเขียนข้อมูล (อย่างใดอย่างหนึ่ง)

2.6.2.2.4 การเขียนข้อมูลลงในหน่วยความจำที่มีการป้องกัน (Write Protection Memory)

คือ การเขียนข้อมูลลงยังแอดเดรสของหน่วยความจำใด ๆ ใน 32 ไบต์แรก คำสั่งนี้มีเงื่อนไขว่าข้อมูลที่เขียนลงไปจะถูกเขียนลงยังแอดเดรสของหน่วยความจำที่กำหนดอย่างถาวร ไม่สามารถแก้ไขเปลี่ยนแปลงอะไรได้อีก สำหรับรูปสัญญาณของกระบวนการนี้อ้างอิงได้จากรูปสัญญาณของการเขียนข้อมูลลงในหน่วยความจำหลัก (Update Main Memory)

Command: WRITE PROTECTION MEMORY

	Control								Address	Data
	B7	B6	B5	B4	B3	B2	B1	B0	A7...A0	D7...D0
Binary	0	0	1	1	1	1	0	0	Address	Input data
Hexadecimal	$3C_H$								00_H...$1F_H$	Input data

ตารางที่ 2.8 รูปแบบคำสั่งในการเขียนข้อมูลลงในหน่วยความจำที่มีการป้องกัน

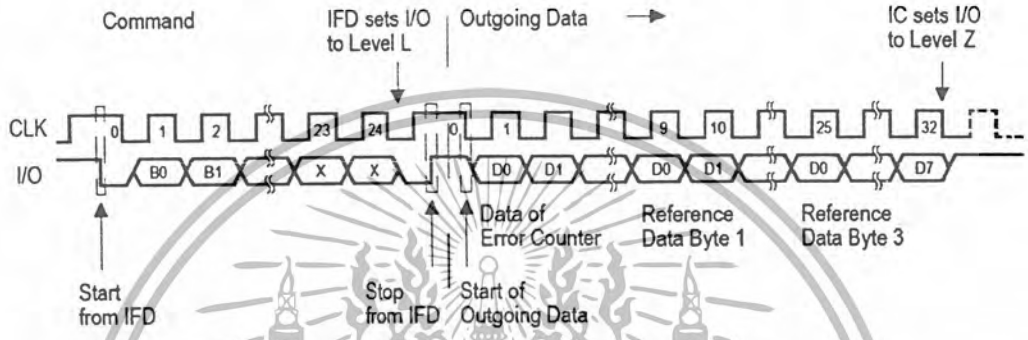
2.6.2.2.5 การอ่านข้อมูลจากหน่วยความจำปลอดภัย (Read Security Memory)

คือ การอ่านค่าของ Error Counter เพื่อตรวจดูว่าการ์ดใบนั้น ๆ ได้ถูกล็อคไปแล้วหรือยัง โดยค่าภายในบิต D2, D1 และ D0 ของ Error Counter จะเป็นส่วนที่บอกถึงสถานะของการ์ดใบนั้น ๆ หากค่าของบิต D2, D1 และ D0 เป็น 0 ทั้งหมด ก็แสดงว่าการ์ดได้ถูกล็อคไปแล้ว ซึ่งจะไม่สามารถแก้ไขอะไรได้และจะไม่สามารถเขียนข้อมูลลงยังการ์ดนั้นได้อีกต่อไป (แต่ว่าการอ่านข้อมูลในการ์ดจะยังคงทำได้ตามปกติ)

Command: READ SECURITY MEMORY

	Control								Address	Data
	B7	B6	B5	B4	B3	B2	B1	B0	A7...A0	D7...D0
Binary	0	0	1	1	0	0	0	1	No effect	No effect
Hexadecimal	31_H								No effect	No effect

ตารางที่ 2.9 รูปแบบคำสั่งในการอ่านข้อมูลจากหน่วยความจำปลอดภัย



รูปที่ 2-14 รูปสัญญาณของการอ่านข้อมูลจากหน่วยความจำปลอดภัย

2.6.2.2.6 การเขียนข้อมูลลงในหน่วยความจำปลอดภัย (Update Security Memory)

คือ การเข้าไปแก้ไขข้อมูลของรหัส PSC ภายในการ์ด หรืออาจกล่าวได้ว่าเป็นการเข้าไปเปลี่ยนรหัสป้องกันของการ์ดนั้นเอง คำสั่งจะถูกกระทำต่อเมื่อมีการส่งรหัส PSC ที่ถูกต้องไปยังการ์ดเสียก่อน โดยในกรณีที่ป้อนรหัสผิด ค่าของบิต D2, D1 และ D0 ใน Error Counter จะค่อย ๆ ถูกเปลี่ยนจากค่า "1" เป็น "0" ไล่ไปทีละบิตตามจำนวนครั้งที่ป้อนผิด หากทั้งหมดกลายเป็นศูนย์เมื่อไรการ์ดก็จะถูกล็อคทันที ซึ่งนั่นหมายความว่าโอกาสที่ป้อนรหัสผิดมีเพียง 3 ครั้งเท่านั้น สำหรับรูปสัญญาณของกระบวนการนี้ จะเหมือนกับรูปสัญญาณของการเขียนข้อมูลลงในหน่วยความจำหลัก

Command: UPDATE SECURITY MEMORY

	Control								Address	Data
	B7	B6	B5	B4	B3	B2	B1	B0	A7...A0	D7...D0
Binary	0	0	1	1	1	0	0	1	Address	Input data
Hexadecimal	39_H								$00_H...03_H$	Input data

ตารางที่ 2.10 รูปแบบคำสั่งในการเขียนข้อมูลลงในหน่วยความจำปลอดภัย

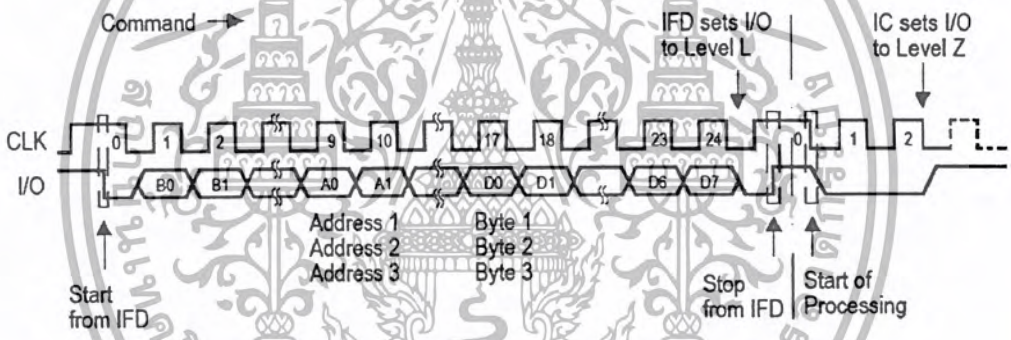
2.6.2.2.7 การเปรียบเทียบและพิสูจน์ข้อมูล (Compare Verification Data)

คือ การสั่งให้การ์ดทำการเปรียบเทียบรหัส PSC กับรหัสผ่านที่เราได้ส่งไปยังการ์ด ในการเปรียบเทียบที่ว่านี้ ข้อมูลที่การ์ดจะส่งกลับมาคือค่าของ Error Counter ที่จะบอกว่ารหัสที่เราป้อนนั้นถูกต้องหรือไม่ และยังเหลือโอกาสพลาดอีกกี่ครั้งเท่านั้น (โดยเราจะไม่สามารถเข้าไปอ่านรหัส PSC ของการ์ดออกมาได้)

Command: COMPARE VERIFICATION DATA

	Control								Address	Data
	B7	B6	B5	B4	B3	B2	B1	B0	A7…A0	D7…D0
Binary	0	0	1	1	0	0	1	1	Address	Input data
Hexadecimal	33_H								$00_H…03_H$	Input data

ตารางที่ 2.11 รูปแบบคำสั่งในการเปรียบเทียบและพิสูจน์ข้อมูล



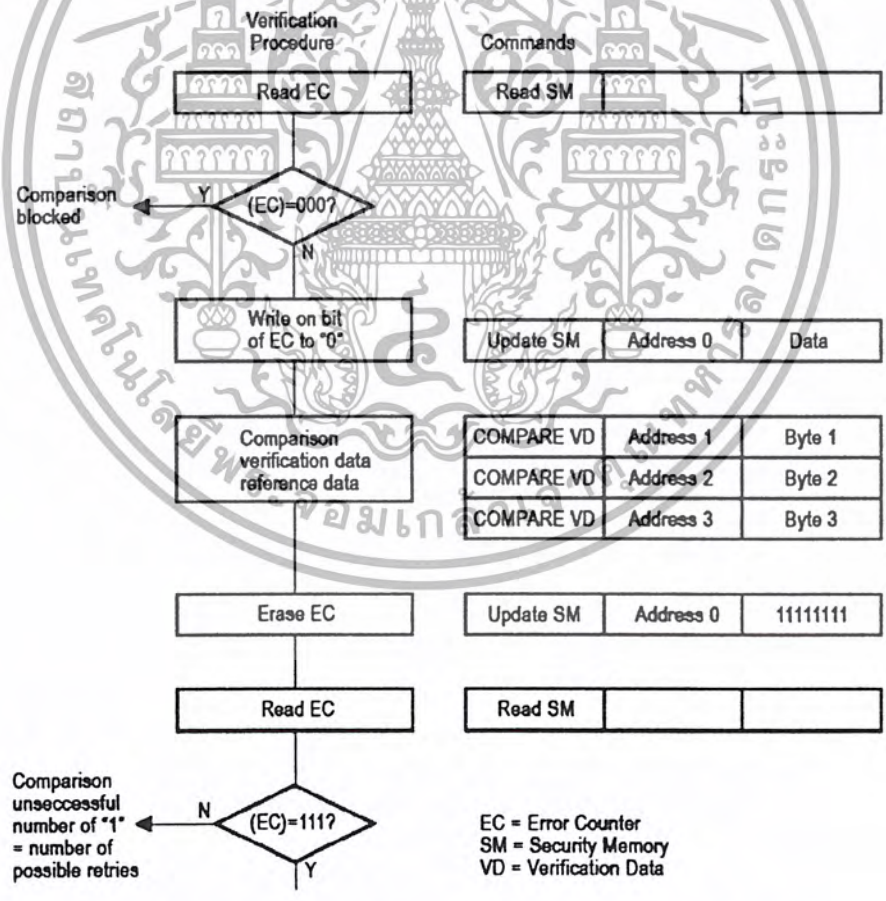
รูปที่ 2-15 รูปสัญญาณของการเปรียบเทียบและพิสูจน์ข้อมูล

การเปรียบเทียบค่า PSC

ในสมาร์ตการ์ด เบอร์ SLE 4442 ผลลัพธ์ที่ได้จากการเปรียบเทียบค่า PSC ที่ถูกเก็บอยู่ในหน่วยความจำที่มีระบบรักษาความปลอดภัย ต้องถูกต้องเพื่อที่จะสามารถทำการเปลี่ยนแปลงหรือแก้ไขข้อมูล เมื่อเราทำการป้อนรหัส PSC ผิดนั้นจะเป็นผลทำให้บิตถูกเปลี่ยนจากลอจิกสูงไปสู่ลอจิกต่ำ ซึ่งไม่สามารถเปลี่ยนกลับเป็นลอจิกสูงได้ ถ้าป้อน PSC ผิด 3 ครั้ง จะทำให้บิตถูกเปลี่ยนครบ 3 ครั้ง ซึ่งจะมีผลทำให้บัตรสมาร์ตการ์ดใบนั้นไม่สามารถลบ และเขียนข้อมูลได้อีก แต่ยังคงอ่านข้อมูลได้ตามปกติ

Command	Control	Address	Data	Remark
	B7…B0	A7…A0	D7…D0	
Read Security Memory	31_H	No effect	No effect	Check Error Counter
Update Security Memory	39_H	00_H	Input Data	Write free bit in Error Counter input data: 0000 0ddd binary
Compare Verification Data	33_H	01_H	Input Data	Reference Data Byte 1
Compare Verification Data	33_H	02_H	Input Data	Reference Data Byte 1
Compare Verification Data	33_H	03_H	Input Data	Reference Data Byte 1
Update Security Memory	39_H	00_H	FF_H	Erase Error Counter
Read Security Memory	31_H	No effect	No effect	Check Error Counter

ตารางที่ 2.12 แสดงรูปแบบคำสั่ง PSC ในการเข้าถึงหน่วยความจำแบบต่างๆ



รูปที่ 2-16 แสดงกระบวนการเปรียบเทียบรหัสผ่านกับรหัส PSC

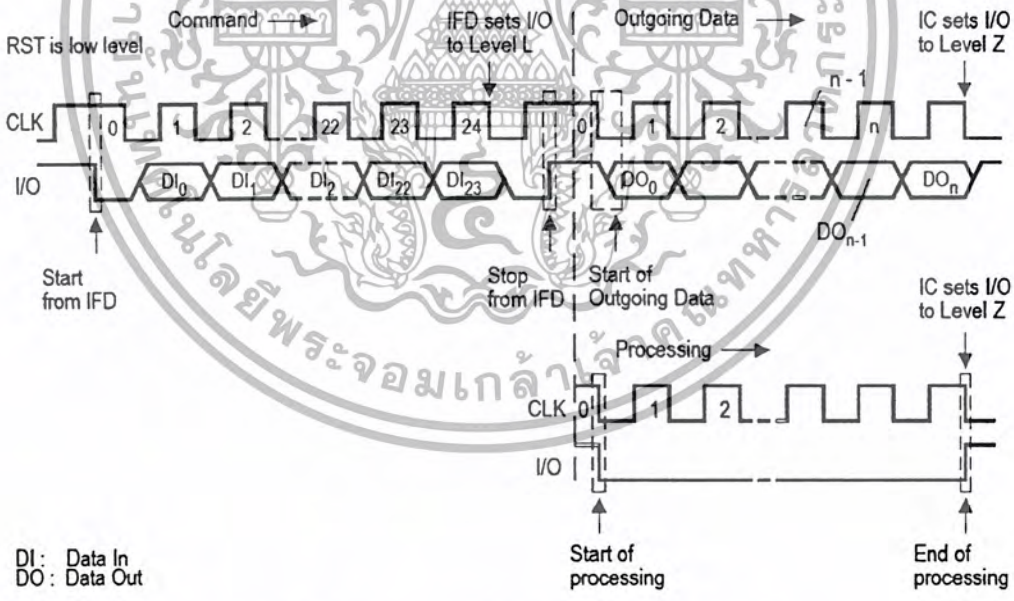
2.6.2.3 โหมดการอ่านข้อมูล (Outgoing Data Mode)

โหมดการทำงานนี้จะเกิดขึ้นหลังจากที่มีการส่งคำสั่งในกลุ่มของการขออ่านข้อมูลไปยังสมาร์ตการ์ดเพื่อขออ่านข้อมูลจากพื้นที่ใด ๆ ในหน่วยความจำ หลังจากที่ได้รับคำสั่งดังกล่าว สมาร์ตการ์ดจะส่งข้อมูลที่ถูกร้องขอกลับมายังเครื่องอ่าน ซึ่งก็เท่ากับว่าเครื่องอ่านจะสามารถอ่านข้อมูลที่ต้องการออกมาได้สำเร็จจากโหมดการทำงานนี้

2.6.2.4 โหมดดำเนินการ (Processing Mode)

โหมดดำเนินการจะเกิดขึ้นหลังจากที่มีการส่งคำสั่งในกลุ่มของการขอเขียนหรือลบข้อมูลออกจากพื้นที่ใด ๆ ในหน่วยความจำ โดยหลังจากที่ได้รับคำสั่งดังกล่าว สมาร์ตการ์ดจะเริ่มดำเนินกระบวนการตามที่ได้รับการสั่งมา ในโหมดการทำงานนี้ข้อมูลจากขา I/O จะไม่ถูกนำมาใช้ร่วมในการทำงานเลย (โดยจะมีสถานะเป็นลอจิกต่ำตลอดทั้งช่วง)

โหมดการส่งคำสั่ง, โหมดการอ่านข้อมูล และโหมดการดำเนินการ จะเรียกรวมกันว่าเป็นโหมดการประมวลผล (Operational Modes) ซึ่งรูปสัญญาณของโหมดการประมวลผล ดังแสดงในรูปที่ 2-17



รูปที่ 2-17 รูปสัญญาณของโหมดการประมวลผล

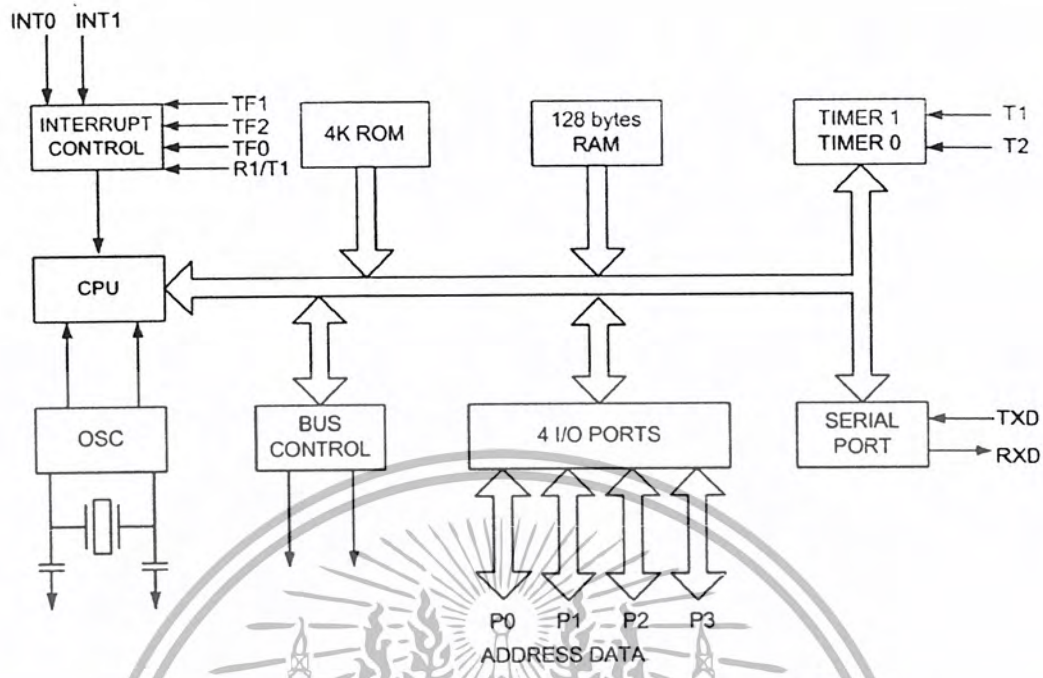
2.7 ไมโครคอนโทรลเลอร์ MCS-51

ไมโครคอนโทรลเลอร์มาจากคำ 2 คำรวมกันคือ "ไมโคร" (micro) ซึ่งหมายถึง ไมโครโปรเซสเซอร์ (microprocessor) ซึ่งเป็นอุปกรณ์ประมวลผลข้อมูลขนาดเล็ก ภายในประกอบด้วย หน่วยประมวลผลกลางหรือซีพียู (CPU: Central Processing Unit) หน่วยคำนวณทางคณิตศาสตร์และลอจิก (ALU: Arithmetic Logic Unit) วงจรเชื่อมต่อหน่วยความจำ และวงจรสัญญาณนาฬิกา อีกคำหนึ่งคือคำว่า "คอนโทรลเลอร์" (controller) หมายถึงอุปกรณ์ควบคุม ดังนั้น ไมโครคอนโทรลเลอร์จึงเป็นอุปกรณ์ที่ใช้ในการควบคุมโดยที่สามารถเขียนโปรแกรมเพื่อกำหนดรูปแบบการควบคุมได้อย่างอิสระ

ไมโครคอนโทรลเลอร์ ตระกูล MCS-51 มีด้วยกันหลายเบอร์ ขึ้นอยู่กันโครงสร้างภายใน เช่น บางเบอร์มีหน่วยความจำแบบ ROM บางเบอร์เป็นแบบ EPROM หรือบางเบอร์ไม่มีหน่วยความจำภายในเป็นต้น อย่างไรก็ตามลักษณะขาต่าง ๆ จะเหมือนกัน โดยในโครงงานนี้จะใช้ไมโครคอนโทรลเลอร์ เบอร์ MCS-51 (IC AT89C51) ซึ่งมีหน่วยความจำโปรแกรมเป็นแบบ EPROM

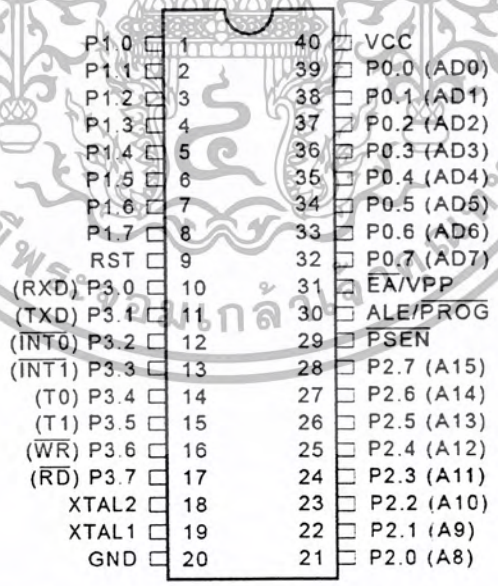
2.7.1 คุณลักษณะพื้นฐานของ MCS-51

- เป็นหน่วยประมวลผลกลางขนาด 8 บิต
- มีความสามารถประมวลผลของลอจิกระดับบิต
- มีขนาดของหน่วยความจำสำหรับโปรแกรมทำงานได้ถึง 64 กิโลไบต์ (Program Memory)
- มีขนาดของหน่วยความจำสำหรับเก็บข้อมูลได้ถึง 64 กิโลไบต์ (Data Memory)
- มีหน่วยความจำสำหรับโปรแกรมภายในขนาด 4 กิโลไบต์
- มีหน่วยความจำข้อมูลภายในขนาด 128 ไบต์
- มีพอร์ตสำหรับควบคุม 4 พอร์ต สามารถอ้างอิงพอร์ตได้ระดับบิตต่อบิต
- มีชุด Timer/Counter ขนาด 16 บิต 2 ชุด ทำงานได้ 4 โหมด
- มีพอร์ตรับส่งข้อมูลอนุกรม (UART) 2 พอร์ต แบบ Full Duplex เลือกรูปแบบได้ 4 โหมด
- มีวงจรควบคุมการอินเตอร์รัปต์จากแหล่งกำเนิดสัญญาณได้ 6 ประเภท
- มีวงจรออสซิลเลเตอร์ภายใน



รูปที่ 2-18 แสดงโครงสร้างภายในของไมโครคอนโทรลเลอร์ 8051

2.7.2 การจัดขามาตรฐานของของไมโครคอนโทรลเลอร์ MCS-51



รูปที่ 2-19 การจัดขามาตรฐานของของไมโครคอนโทรลเลอร์ MCS-51

ไมโครคอนโทรลเลอร์ MCS-51 ทุกเบอร์จะมีขาใช้งานพื้นฐานเหมือนกัน ดังแสดงในรูปที่ 2-19 โดยมีรายละเอียดขั้นต้น ดังนี้

ขา Vcc ใช้สำหรับต่อไฟเลี้ยง +5 โวลต์

ขา GND เป็นขากราวด์ สำหรับต่อกับกราวด์ของระบบ

ขาพอร์ต 0 (P0.0-P0.7) มี 8 ขา แต่ละขาสามารถกำหนดให้เป็นได้ทั้งอินพุตและเอาท์พุต สำหรับใช้งานทั่วไป ถ้าหากต้องการกำหนดให้ขาพอร์ต 0 ขาใดขาหนึ่งเป็นอินพุต สามารถทำได้โดยการเขียนข้อมูล "1" ไปยังแต่ละบิตของพอร์ตที่ต้องการติดต่อด้วย ส่งผลให้ขานั้นมีสถานะปล่อยลอย (float) จึงมีอินพุตอิมพีแดนซ์สูง สามารถใช้งานเป็นขาพอร์ตอินพุตได้ นอกจากนั้นขาพอร์ตนี้ยังถูกใช้งานในการติดต่อกับขาแอดเดรสไบต์ต่ำของหน่วยความจำภายนอก (A0-A7) และขาข้อมูล (D0-D7) โดยใช้กระบวนการมัลติเพล็กซ์เข้าช่วย เพื่อสลับการทำงานให้เป็นได้ทั้งขาติดต่อแอดเดรสและขาข้อมูล

ขาพอร์ต 1 (P1.0-P1.7) มี 8 ขา แต่ละขาสามารถกำหนดให้เป็นได้ทั้งอินพุตและเอาท์พุต สำหรับใช้งานทั่วไป นอกจากนี้ในอนุกรม AT89Sxx จะใช้ขา P1.0 เป็นขาอินพุตสำหรับนับค่าของไทม์เมอร์ 2 และ P1.1 เป็นขาอินพุตทริกเกอร์ของไทม์เมอร์ 2 ในขณะที่ขา P1.4 ถึง P1.7 เป็นขาสำหรับเชื่อมต่อแบบ SPI เพื่อทำการโปรแกรมข้อมูลในระบบ

ขาพอร์ต 2 (P2.0-P2.7) มี 8 ขา แต่ละขาสามารถกำหนดให้เป็นได้ทั้งอินพุตและเอาท์พุต สำหรับใช้งานทั่วไป นอกจากนั้นขาพอร์ตนี้ยังถูกใช้งานในการติดต่อกับขาแอดเดรสไบต์สูงของหน่วยความจำภายนอก (A8-A15)

ขาพอร์ต 3 (P3.0-P3.7) มี 8 ขา แต่ละขาสามารถกำหนดให้เป็นได้ทั้งอินพุตและเอาท์พุต สำหรับใช้งานทั่วไป นอกจากนั้นขาพอร์ต 3 ยังเป็นขาที่มีหน้าที่การใช้งานพิเศษ ดังแสดงในตารางที่ 2.13

ขาพอร์ต	หน้าที่
P3.0	RxD – ใช้เป็นขาอินพุตสำหรับรับข้อมูลจากการสื่อสารแบบอนุกรม
P3.1	TxD – ใช้เป็นขาอินพุตสำหรับส่งข้อมูลจากการสื่อสารแบบอนุกรม
P3.2	INT0 - ใช้เป็นขาอินพุตสำหรับรับสัญญาณอินเตอร์รัปต์จากภายนอกช่องที่ 0
P3.3	INT1 - ใช้เป็นขาอินพุตสำหรับรับสัญญาณอินเตอร์รัปต์จากภายนอกช่องที่ 1
P3.4	T0 - ใช้เป็นขาอินพุตสำหรับรับสัญญาณไทเมอร์จากภายนอกช่องที่ 0
P3.5	T1 - ใช้เป็นขาอินพุตสำหรับรับสัญญาณไทเมอร์จากภายนอกช่องที่ 1
P3.6	WR – ใช้เป็นขาสัญญาณควบคุมการเขียนข้อมูลไปยังหน่วยความจำภายนอก
P3.7	RD – ใช้เป็นขาสัญญาณควบคุมการอ่านข้อมูลจากหน่วยความจำภายนอก

ตารางที่ 2.13 หน้าที่พิเศษของขาพอร์ต 3

ขารีเซต ใช้ในการรีเซตการทำงานของไมโครคอนโทรลเลอร์ โดยในการป้อนสัญญาณเพื่อรีเซต สถานะที่ขานี้ต้องอยู่ในระดับรีเซตอย่างน้อย 2 แมชชีนไซเคิล

ขา ALE/PROG (Address Latch Enable/Program pulse input) เป็นขาที่ใช้ในการควบคุมการแลตช์ของขาพอร์ต 0 เมื่อมีการใช้งานหน่วยความจำภายนอก นอกจากนั้นขานี้ยังใช้เป็นขาสำหรับรับพัลส์ของการโปรแกรมสำหรับโปรแกรมข้อมูลลงในไมโครคอนโทรลเลอร์ MCS-51 ในรุ่นที่มีหน่วยความจำโปรแกรมเป็นแบบ EEPROM

ขา PSEN (Program Store Enable) ขานี้ใช้ในการส่งสัญญาณเพื่อร้องขอติดต่อกับหน่วยความจำโปรแกรมภายนอก เมื่อไมโครคอนโทรลเลอร์ต้องการอ่านข้อมูลจากหน่วยความจำโปรแกรมภายนอกตัวไมโครคอนโทรลเลอร์จะส่งสัญญาณออกมาที่ขานี้ 2 ครั้งในแต่ละแมชีนไซเคิล แต่ถ้าหากติดต่อกับหน่วยความจำข้อมูลภายนอก ขานี้จะไม่มีการส่งสัญญาณใด ๆ ออกมา

ขา EA/Vpp (External Access enable/Programming voltage input) ใช้สำหรับเลือกการติดต่อหน่วยความจำโปรแกรมจากภายนอกหรือภายในตัวไมโครคอนโทรลเลอร์ ถ้าหากขานี้เป็น "0" เป็นการเลือกให้ไมโครคอนโทรลเลอร์ติดต่อกับหน่วยความจำโปรแกรมภายนอก แต่ถ้าหากขานี้เป็น "1" เป็นการเลือกให้ไมโครคอนโทรลเลอร์ติดต่อกับหน่วยความจำภายในตัวไมโครคอนโทรลเลอร์ นอกจากนี้ ที่ขานี้ยังใช้เป็นขาอินพุตสำหรับแรงดันไฟสูงสำหรับการโปรแกรมหน่วยความจำภายในไมโครคอนโทรลเลอร์ สำหรับในไมโครคอนโทรลเลอร์ MCS-51 แบบเฟลชต้องการแรงดันสำหรับการโปรแกรม +12 โวลต์

ขา XTAL1 และ XTAL2 เป็นขาสำหรับต่อคริสตอลเพื่อสร้างสัญญาณนาฬิกาในการกำหนดจังหวะการทำงานของไมโครคอนโทรลเลอร์

2.7.3 พอร์ตอนุกรมของไมโครคอนโทรลเลอร์ MCS-51

ไมโครคอนโทรลเลอร์ทั้งหมดในตระกูล MCS-51 เป็นไมโครคอนโทรลเลอร์ที่มีวงจรสื่อสารแบบฟูลดูเพล็กซ์ คือมีการเชื่อมโยงข้อมูลในลักษณะสองทิศทางได้ในเวลาเดียวกัน ทำให้รับและส่งข้อมูลได้พร้อมกัน โดยวงจรสื่อสารข้อมูลแบบอนุกรมของไมโครคอนโทรลเลอร์ MCS-51 นี้เป็นแบบอะซิงโครนัส ซึ่งตัวรับส่งข้อมูลชนิดอะซิงโครนัส จะมีบัฟเฟอร์สำหรับข้อมูลเป็นพิเศษเพื่อเพิ่มความเร็วในการสื่อสาร พอร์ตชนิดอนุกรมนั้นสามารถเลือกโปรแกรมเพื่อเลือกใช้การทำงานแบบใดแบบหนึ่งใน 4 แบบ ด้วยการใช้โปรแกรมควบคุมอัตราการส่งข้อมูลและรูปแบบของข้อมูล อัตราการส่งข้อมูลที่เลือกใช้ได้สูงถึง 19,200 บิต/วินาที ด้วยความเร็วของสัญญาณนาฬิกา 1 MHz สำหรับใช้ในระบบเครือข่าย (Networks) และระบบการสื่อสารของไมโครคอนโทรลเลอร์หลายตัวร่วมกัน จะเลือกความเร็วของสัญญาณนาฬิกาด้วยวงจรนับและวงจรตั้งเวลา

2.8 การใช้ LCD Module

ใน LCD Module จะมีส่วนประกอบหลัก ๆ 3 ส่วน ดังนี้

1. ตัวแสดงผล (display) ภายในเป็นผลึกเหลวที่สามารถแสดงผลให้เห็นโดยอาศัยแสงจากภายนอก ดังนั้นจึงต้องมีมุมในการมองข้อมูลที่แสดงผลบนจอ LCD
2. ตัวควบคุม (controller) เป็นตัวรับข้อมูลจากอุปกรณ์ภายนอกมาควบคุมการทำงานของ LCD Module เช่น ลบจอภาพ แสดงตัวอักษร หรือเลื่อนเคอร์เซอร์ เป็นต้น
3. ตัวขับ (driver) เป็นตัวรับสัญญาณจากตัวควบคุมมาขับให้ตัวแสดงผลแสดงข้อมูลตามที่กำหนด

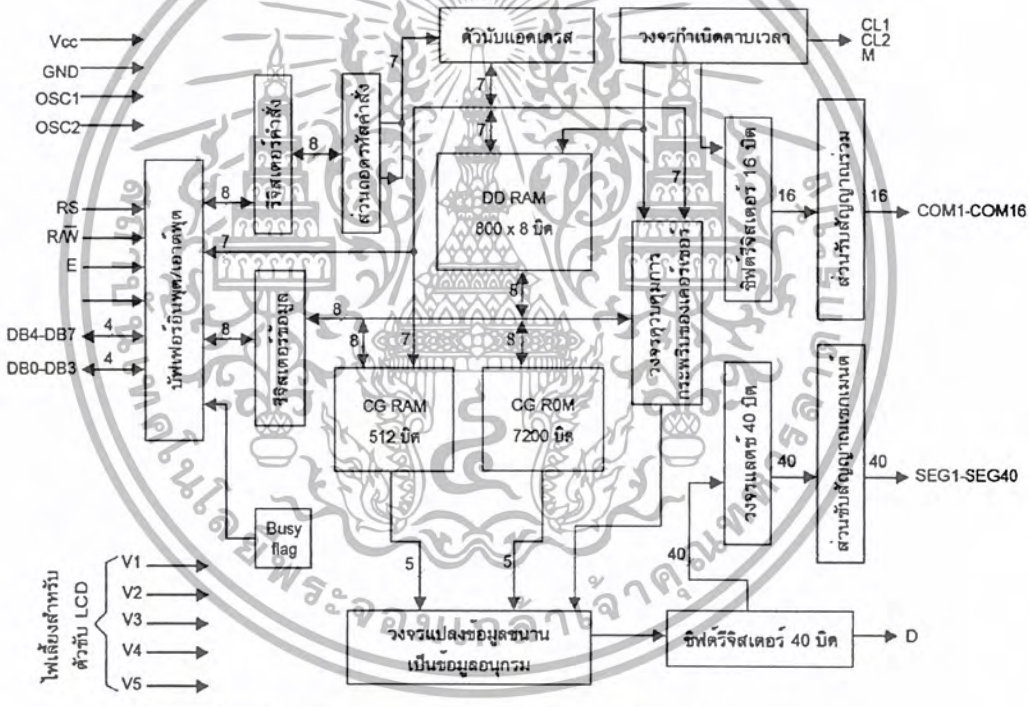
LCD Module แบ่งได้เป็น 2 แบบคือ แบบ Dot matrix และ Graphic โดยแบบ Dot matrix จะแสดงผลเป็นตัวอักษรขนาด 5x8 Dot มีจำนวนอักษรและบรรทัดแตกต่างกันในแต่ละรุ่น ส่วนแบบ Graphic จะแสดงผลในแบบ Bit map คือ สามารถสร้างเป็นภาพใด ๆ ก็ได้ตามต้องการ ซึ่งโดยทั่วไปมักจะใช้แบบ Dot matrix มากกว่า เนื่องจากราคาถูกกว่าและเพียงพอต่องานส่วนใหญ่

คุณสมบัติของ Dot matrix LCD Module สามารถสรุปเป็นข้อ ๆ ดังนี้

1. มีให้เลือกหลายรุ่นตามแต่ต้องการใช้งาน โดยมีจำนวนตัวอักษรและบรรทัดแตกต่างกัน
2. ตัวอักษรแสดงด้วย Dot matrix ขนาด 5x8 Dot.
3. สามารถต่อเข้ากับระบบไมโครคอนโทรลเลอร์ได้ 2 ลักษณะคือ แบบ Memory map และแบบผ่าน 8255 พอร์ต ซึ่งจะใช้ขาสัญญาณทั้งหมด 14 ขา

4. การใช้งานง่ายและสะดวก ระบบไมโครคอนโทรลเลอร์เพียงแต่ส่งข้อมูลให้กับ LCD Module เท่านั้น ข้อความก็จะปรากฏและค้างไว้บนแผงแสดง ทำให้ไม่ต้องเสียเวลาของระบบ
5. มีคำสั่งพิเศษสำหรับอำนวยความสะดวกเป็นจำนวนมาก เช่น CLEAR, DISPLAY, HOME, CURSOR ON/OFF เป็นต้น
6. สามารถแสดงผลเป็นตัวอักษรภาษาอังกฤษและตัวเลขได้ 160 ตัว และสัญญาณพิเศษอีก 32 ตัว รวมทั้งสามารถกำหนดอักษรที่ออกแบบเองได้อีก 8 ตัว
7. กินกระแสและมีน้ำหนักน้อย รวมทั้งทำงานได้ด้วยไฟเลี้ยงระดับ 5 โวลต์เท่านั้น

2.8.1 โครงสร้างภายในของตัวควบคุม LCD Module



รูปที่ 2-20 ไดอะแกรมการทำงานของ LCD Module แบบอักขระ

จากรูปที่ 2-20 เป็นบล็อกไดอะแกรมภายในของชิปควบคุม LCD เบอร์ HD44780 ซึ่งเป็น LCD Module แบบอักขระ ซึ่งประกอบด้วย

- บัฟเฟอร์อินพุตเอาท์พุต เป็นส่วนที่ใช้ในการติดต่อรับส่งข้อมูลกับอุปกรณ์ภายนอก เพื่อที่จะถ่ายทอดข้อมูลเข้าออกภายในตัวควบคุม

- รีจีสเตอร์คำสั่ง (Instruction Register: IR) เป็นรีจีสเตอร์ที่ใช้รับข้อมูลคำสั่งจากอุปกรณ์

ภายนอก เพื่อนำไปควบคุมการแสดงผล

- รีจีสเตอร์ข้อมูล (Data Register: DR) เป็นรีจีสเตอร์ใช้รับข้อมูลจากอุปกรณ์ภายนอก เพื่อส่งต่อไปยังหน่วยความจำที่ทำหน้าที่เก็บข้อมูลแสดงผล หรือนำข้อมูลไปสร้างตัวอักษรเพิ่มเติมในแรมเก็บตัวอักษร

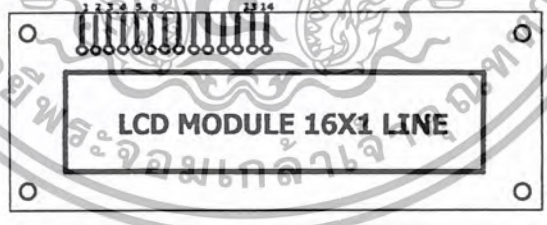
- แรมเก็บข้อมูลแสดงผล (Display Data RAM: DDRAM) เป็นหน่วยความจำแรมทำหน้าที่เก็บข้อมูลที่มาจากรีจีสเตอร์ DR ตัวควบคุมจะนำข้อมูลใน DDRAM นี้ไปเปิดตาราง (Look up-label) ของตัวอักษรที่เก็บไว้ในหน่วยความจำรอมและแรมเก็บตัวอักษร เพื่อนำไปแสดงที่ตัวแสดงผล

- รอมเก็บตัวอักษร (Character Generator ROM: CGROM) เป็นหน่วยความจำรอมที่ใช้เก็บข้อมูลตัวอักษรหรือสัญลักษณ์ที่สามารถอ่านออกไปแสดงที่ตัวแสดงผลได้ มีขนาด 7200 บิต โดยจะถูกอ่านด้วยค่าของข้อมูลใน DDRAM

- แรมเก็บตัวอักษร (Character Generator RAM: CGRAM) เป็นหน่วยความจำแรมที่ใช้เก็บอักษรที่มีการสร้างเพิ่มเติมขึ้นใหม่ ในกรณีที่ตัวอักษรใน CGROM ไม่เพียงพอ มีขนาด 512 บิต การเขียนและอ่านค่าไปใช้นั้นทำได้เช่นเดียวกับ CGROM คือ เขียนข้อมูลลงใน DDRAM แล้วตัวควบคุมจะมาอ่านค่าจาก CGRAM เอง

2.8.2 LCD Module ขนาด 16 ตัวอักษร 1 บรรทัด (LCD 16x1)

สำหรับ LCD Module ที่ใช้ในโครงงานนี้ เป็นแบบ Dot Matrix มีขนาด 16 ตัวอักษร 1 บรรทัด เนื่องจากราคาถูก หาง่าย และเป็น LCD Module ที่มีโครงสร้างที่เป็นมาตรฐาน



รูปที่ 2-21 รูปร่างและการจัดขา LCD Module แบบอักขระ

LCD Module ขนาด 16x1 มีขาต่อใช้งานทั้งสิ้น 14 ขา มีการจัดขาดังแสดงในรูปที่ 2-21 สำหรับรายละเอียดการทำงานของแต่ละขามีดังนี้

V_{ss} (ขา 1): ต่อกราวด์

V_{DD} (ขา 2): ต่อไฟเลี้ยง +5 โวลต์

V_0 (ขา 3): เป็นขาอินพุตรับแรงดันเพื่อปรับความเข้มของการแสดงผล

RS (ขา 4): เป็นขาอินพุตใช้ในการแยกชนิดของข้อมูลที่ทำการประมวลผลในขณะนั้นว่าเป็นคำสั่งสำหรับรีจีสเตอร์ IR หรือเป็นข้อมูลสำหรับรีจีสเตอร์ DR โดยถ้าขานี้เป็น "0" ข้อมูลที่ส่งมาจะเป็นคำสั่ง แต่ถ้าขานี้เป็น "1" ข้อมูลที่ส่งมาจะเป็นข้อมูลสำหรับการแสดงผล

R/W (ขา 5): เป็นขาที่ใช้เลือกการอ่านหรือเขียนข้อมูลกับ LCD ถ้าเป็น "0" เป็นการกำหนดให้เขียนข้อมูล แต่ถ้าเป็น "1" จะเป็นการอ่านข้อมูล

E (ขา 6): เป็นขาอีนาเบิล LCD ให้ทำงาน

D0-D7 (ขา 7-14): ใช้เป็นทางผ่านของข้อมูลระหว่าง LCD กับอุปกรณ์ภายนอกขนาด 8 บิต

2.8.3 คำสั่งควบคุม LCD Module

Instruction	RS	R/W	Data Bit								Execute Time (nS)
			7	6	5	4	3	2	1	0	
Clear Display	0	0	0	0	0	0	0	0	0	1	1640
Cursor At Home	0	0	0	0	0	0	0	0	1	*	1640
Entry Mode Set	0	0	0	0	0	0	0	1	I/D	S	40
Display On/Off	0	0	0	0	0	0	1	D	C	B	40
Display Shift	0	0	0	0	0	1	S/C	R/L	*	*	40
Function Set	0	0	0	0	1	DL	N	F	*	*	40
Set CGRAM Address	0	0	0	1	CGRAM Address						40
Set DDRAM Address	0	0	1	DDRAM Address							40
Busy, Address Read	0	1	BF	Address							0
CGRAM, DDRAM WR	1	0	Write Data								40
CGRAM, DDRAM RD	1	1	Read Data								40

ตารางที่ 2.14 แสดงชุดคำสั่งเวลาและที่ LCD Module ใช้ในการทำงานแต่ละคำสั่ง

1. คำสั่งเคลียร์ตัวแสดงผล (Clear Display)

มีข้อมูลคำสั่งเป็น 01H เป็นคำสั่งที่ใช้เขียนข้อมูลช่องว่าง หรือ space เข้าไปใน DDRAM ทั้งหมด เมื่อตัวควบคุมเอ็กซิคิวต์คำสั่งนี้ จะกำหนดแอดเดรสของ DDRAM เป็น 0 เคอร์เซอร์จะกลับไปอยู่ตำแหน่งซ้ายมือสุดของจอแสดงผล แล้วเซตบิต I/D (ซึ่งจะกล่าวถึงภายหลัง) ให้เป็น "1"

2. คำสั่ง Cursor at Home (หรือ Return Home)

ต้องกำหนดให้บิต 1 ของข้อมูลเป็น "1" เป็นคำสั่งให้เคอร์เซอร์เคลื่อนที่กลับไปยังตำแหน่งซ้ายสุดของจอแสดงผล แต่ข้อมูลบนจอแสดงผลไม่เปลี่ยนแปลง นั่นคือ ข้อมูลคำสั่งของคำสั่งนี้จะเป็น 02H หรือ 03H ก็ได้

3. คำสั่งเลือกโหมดการป้อนข้อมูล (Entry mode Set)

ใช้สำหรับการกำหนดการเลื่อนของเคอร์เซอร์และตำแหน่งแอดเดรสของ DDRAM ดังนี้

- บิต S เป็นบิตที่ใช้ในการกำหนดลักษณะของการแสดงผล เมื่อมีการป้อนข้อมูล ถ้าหากบิต S เป็น "1" เมื่อเกิดข้อมูลใหม่บนจอแสดงผล ตัวเคอร์เซอร์จะอยู่กับที่ แต่ตัวอักษรข้อมูลเดิมจะถูกดันไปทางซ้าย แต่ถ้าหากบิตนี้เป็น "0" เมื่อเกิดข้อมูลใหม่ตัวเคอร์เซอร์จะเลื่อนไปทางขวามือ

- บิต I/D เป็นบิตที่ใช้ในการกำหนดว่าเมื่อเขียนหรืออ่านข้อมูลแล้ว ทำให้แอดเดรสของ DDRAM เพิ่มขึ้นหรือลดลงหนึ่งแอดเดรส โดยถ้าบิตนี้เป็น "1" แอดเดรสของ DDRAM จะเพิ่มขึ้น แต่ถ้าเป็น "0" แอดเดรสจะลดลง

ดังนั้นข้อมูลคำสั่งที่เกิดขึ้นสำหรับคำสั่งนี้ได้แก่ 04H-07H (4 ข้อมูลคำสั่ง) และที่ใช้บ่อยคือ 06H หมายถึง กำหนดให้เมื่อเกิดข้อมูลใหม่ เคอร์เซอร์จะเลื่อนไปทางขวามือ และแอดเดรสของ DDRAM เพิ่มขึ้น

4. คำสั่งควบคุมการแสดงผล (Display ON/OFF)

- บิต D ใช้ควบคุมการเปิดปิดจอแสดงผล ถ้าบิตนี้เป็น "1" จะเป็นการเปิดจอแสดงผล ถ้าเป็น "0" จะเป็นการปิดจอแสดงผล

- บิต C ใช้ควบคุมการแสดงตัวเคอร์เซอร์บนจอแสดงผล ถ้าต้องการให้มีเคอร์เซอร์ ต้องกำหนดให้บิตนี้เป็น "1" ถ้ากำหนดให้เป็น "0" จะเป็นการปิดเคอร์เซอร์ หรือไม่แสดงเคอร์เซอร์

- บิต B ใช้ควบคุมการกระพริบของเคอร์เซอร์ ถ้าบิตนี้เป็น "1" เคอร์เซอร์จะกระพริบ แต่ถ้าบิตนี้เป็น "0" จะไม่มีการกระพริบที่เคอร์เซอร์

ดังนั้นจะมีข้อมูลคำสั่งได้ตั้งแต่ 08H-0FH (8 รูปแบบคำสั่ง) ที่ใช้บ่อยคือ 0CH เป็นการสั่งให้เปิดจอแสดงผล แต่ไม่แสดงเคอร์เซอร์ และ 0FH เป็นการสั่งให้เปิดจอแสดงผล แสดงเคอร์เซอร์ และสั่งให้เคอร์เซอร์กระพริบ

5. คำสั่งควบคุมการเลื่อนเคอร์เซอร์และข้อมูลตัวอักษร (Display Shift)

การควบคุมการเลื่อนเคอร์เซอร์และตัวอักษรบนจอแสดงผลขึ้นอยู่กับการกำหนดบิต S/C และ R/L ซึ่งสามารถสรุปได้ดังนี้

S/C	*R/L*	*ลักษณะการเลื่อน*	*ข้อมูลคำสั่ง*
0	0	เลื่อนเคอร์เซอร์ไปทางซ้าย	10H-13H
0	1	เลื่อนเคอร์เซอร์ไปทางขวา	14H-17H
1	0	เลื่อนตัวอักษรใหม่ไปทางซ้าย	18H-1BH
1	1	เลื่อนตัวอักษรใหม่ไปทางขวา	1CH-1FH

6. คำสั่งกำหนดฟังก์ชันการทำงาน (Function Set)

- บิต DL ใช้กำหนดจำนวนบิตที่ใช้ติดต่อส่งผ่านข้อมูล ถ้าบิตนี้เป็น “0” จะเป็นการติดต่อแบบ 4 บิต แต่ถ้าเป็น “1” จะเป็นแบบ 8 บิต

- บิต N ใช้กำหนดจำนวนบรรทัดของการแสดงผล ถ้าเป็น “0” จะแสดงผล 1 บรรทัด ถ้าเป็น “1” จะแสดงผล 2 บรรทัด ในกรณีที่จอแสดงผลสามารถแสดงได้มากกว่า 2 บรรทัด และต้องการให้แสดงผลมากกว่า 2 บรรทัด ก็กำหนดบิต N นี้ให้เป็น “1”

- บิต F ใช้เลือกความละเอียดของตัวอักษรในการแสดงผล ถ้าบิตนี้เป็น “0” จะเป็นการแสดงผลแบบ 5x8 จุด และถ้าเป็น “1” จะเป็นการแสดงผลแบบ 5x10 จุด

ข้อมูลคำสั่งที่ใช้บ่อยคือ 38H เป็นการกำหนดให้ LCD Module ทำงานในแบบ 8 บิต แสดงผล 2 บรรทัด และเลือกความละเอียดเป็น 5x8 จุด

7. คำสั่งเลือกแอดเดรสของ CGRAM (Set CGRAM Address)

เมื่อต้องการกำหนดแอดเดรสของ CGRAM ต้องกำหนดให้บิต 7 เป็น “0” บิต 6 เป็น “1” ส่วนอีก 6 บิตที่เหลือจะแทนด้วยค่าแอดเดรสของ CGRAM จะต้องกำหนดแอดเดรสด้วยคำสั่งนี้ก่อนที่จะอ่านหรือเขียนข้อมูลให้ CGRAM โดยแอดเดรสของ CGRAM อยู่ระหว่าง 00H-3FH

8. คำสั่งเลือกแอดเดรสของ DDRAM (Set DDRAM Address)

ใช้ในการเลือกแอดเดรสของ DDRAM ก่อนที่จะทำการอ่านหรือเขียนข้อมูล โดยบิต 7 ต้องเป็น “1” และข้อมูลอีก 7 บิตที่เหลือจะเป็นค่าแอดเดรสของ DDRAM ซึ่งแอดเดรสของ DDRAM จะอยู่ระหว่าง 8CH-0FFH ทั้งนี้จำนวนแอดเดรสยังขึ้นกับการกำหนดสถานะที่บิต N ด้วย หากบิต N

เป็น "0" แอดเดรสของ DDRAM จะอยู่ระหว่าง 80H-0CFH และถ้าบิต N เป็น "1" แอดเดรสของ DDRAM จะมี 2 ช่วงคือ 80H-87H และ 0C0H-0C7H

9. คำสั่งอ่านแฟลก BUSY และแอดเดรส (Read BUSY flag & Address)

เป็นคำสั่งที่ใช้อ่านแฟลก BUSY (BF) โดยแฟลกนี้จะเป็นตัวบอกสถานะของตัวควบคุม LCD ว่าพร้อมจะรับข้อมูลอยู่หรือไม่ ถ้าหากบิต BF เป็น "0" แสดงว่าตัวควบคุม LCD พร้อมรับข้อมูลหรือคำสั่ง แต่ถ้าเป็น "1" แสดงว่าขณะนี้ตัวควบคุม LCD ยังอยู่ในกระบวนการทำงานภายในหรือกำลังประมวลผลข้อมูลอยู่ ยังไม่พร้อมรับข้อมูลหรือคำสั่ง

เมื่อต้องการอ่านแฟลกต้องกำหนดให้ขา R/W เป็น "1" ด้วย แต่สัญญาณที่ RS ยังต้องเป็น "0" อยู่ เพราะข้อมูลนี้เป็นข้อมูลคำสั่ง

นอกจากนี้ ยังใช้เป็นคำสั่งอ่านข้อมูลแอดเดรสของ CGRAM และ DDRAM อีกด้วย โดยบิต 0 ถึงบิต 6 เป็นค่าข้อมูลของแอดเดรสที่ต้องการอ่าน

2.8.4 การเขียนคำสั่งและข้อมูลให้แก่ LCD Module

ในการเขียนข้อมูลเพื่อควบคุมให้ LCD Module แสดงผลตามที่ผู้ใช้งานต้องการ ต้องส่งคำสั่ง (instruction) แล้วกำหนดโหมดการทำงานให้แก่ LCD Module ก่อน จากนั้นจึงค่อยส่งข้อมูล (data) ที่ต้องการแสดงผล เนื่องจากบัสข้อมูลของ LCD Module มี 8 เส้น คือ D0-D7 และใช้เป็นทางผ่านของทั้งคำสั่งและข้อมูล ดังนั้นในการส่งคำสั่งและข้อมูลจึงต้องอาศัยการกำหนดสัญญาณลอจิกที่ขา RS ถ้าหากที่ขา RS ได้ลอจิก "0" หมายความว่า ข้อมูลที่ป้อนให้แก่ LCD Module ขณะนั้นเป็นคำสั่ง ในทางตรงข้าม หากขา RS ได้รับลอจิก "1" ข้อมูลที่ป้อนให้ขณะนั้นเป็นข้อมูลที่ใช้ในการแสดงผล

เมื่อต้องการเขียนหรืออ่านข้อมูลลงใน CGRAM และ DDRAM เริ่มต้นต้องกำหนดแอดเดรสที่ต้องการอ่านหรือเขียนก่อน โดยใช้คำสั่งเลือกแอดเดรส จากนั้นให้ขา RS เป็น "1" เพื่อแจ้งให้ตัวควบคุมภายใน LCD Module ทราบว่าข้อมูลที่กำหนดต่อไปนี้เป็นข้อมูลปกติไม่ใช่คำสั่ง

ในกรณีที่ต้องการอ่านข้อมูลต้องกำหนดให้ขา R/W เป็น "1" ข้อมูลขนาด 8 บิต (หรือ 4 บิต) ก็จะปรากฏบนบัสข้อมูล โดยข้อมูลที่อ่านออกมาได้จะเป็นข้อมูลจากแอดเดรสของ CGRAM หรือ DDRAM ตามที่ต้องการ

ในกรณีที่ต้องการเขียนข้อมูล เมื่อกำหนดแอดเดรสและป้อนลอจิก "1" ให้ขา RS แล้ว ต้องกำหนดให้ขา R/W เป็น "0" ข้อมูลที่อยู่บนบัสข้อมูลจะถูกเขียนลงในรีจีสเตอร์ DR จากนั้นจึงถ่ายทอดลงใน DDRAM ต่อไป

2.8.5 จังหวะการทำงานของ LCD Module

ในการติดต่อกับ LCD Module จะต้องมีการหน่วงเวลาหลังจากที่ทำการส่งรหัสคำสั่งหรือข้อมูล เนื่องจากต้องรอให้คอนโทรลเลอร์ภายใน LCD Module แปลความหมายของรหัสคำสั่งและทำงานตามคำสั่งให้เรียบร้อยก่อน จากนั้นจึงจะรับข้อมูลหรือดำเนินการต่อไป

ดังนั้น ในการใช้งาน LCD Module ผู้เขียนโปรแกรมต้องมีโปรแกรมเพื่อหน่วงเวลารอให้ LCD Module พร้อมทำงานด้วย โดยเมื่อเริ่มจ่ายไฟให้แก่ LCD Module ต้องรอประมาณ 10 มิลลิวินาที เพื่อให้ LCD Module ทำการเตรียมความพร้อมหรืออินิเชียล (initial) หลังจากนั้นก็จะกำหนดลอจิกให้แก่ขา RS ของ LCD Module แล้วต้องหน่วงเวลาอีกประมาณ 2 มิลลิวินาทีเพื่อให้คอนโทรลเลอร์ใน LCD Module แปลความหมายของลอจิกที่ขา RS ว่าข้อมูลต่อไปที่จะได้รับนั้นเป็นรหัสคำสั่งหรือเป็นข้อมูลที่ต้องการแสดงผล จากนั้นจะเป็นการส่งข้อมูลมารอที่บัสข้อมูล D0-D7 (กรณีทำงานในโหมด 8 บิต) ขั้นตอนต่อไปจะเป็นการส่งสัญญาณพัลส์ไปที่ขา E เพื่ออีนาเบิล LCD Module ให้รับข้อมูลจากบัสข้อมูลเข้าไป โดยพัลส์ที่ป้อนเข้าที่ขา E ของ LCD Module ต้องเป็นพัลส์ขอบขาขึ้น จากนั้นทำการหน่วงเวลา 2 มิลลิวินาที

ทั้งหมดที่กล่าวมาคือขั้นตอนและจังหวะในการทำงาน 1 รอบของ LCD Module จะเห็นได้ว่ามีโปรแกรมย่อยที่สำคัญอยู่ 3 โปรแกรมย่อยคือ โปรแกรมอินิเชียล LCD, โปรแกรมหน่วงเวลา และ โปรแกรมย่อยการส่งพัลส์เพื่ออีนาเบิล LCD Module

2.9 RS-232C

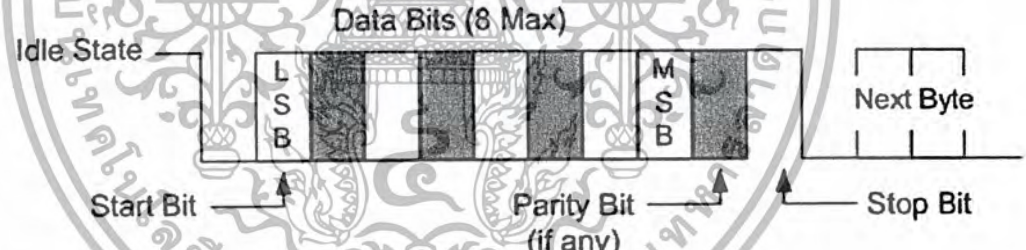
ในปี ค.ศ. 1969 สมาคมผู้ผลิตอุปกรณ์อิเล็กทรอนิกส์ของประเทศสหรัฐอเมริกา (Electronic Industries Association: EIA) ได้กำหนดตามมาตรฐานอุปกรณ์สื่อสารแบบอนุกรม ภายใต้ชื่อว่าพอร์ต RS-232C โดยตัวอักษร RS ย่อมาจาก Recommended Standard และส่วน 232 จะเป็นหมายเลขบ่งบอกของมาตรฐานตัวนี้ ตัวอักษร C คือหมายเลขของฉบับสุดท้ายของมาตรฐาน จุดประสงค์ของมาตรฐานตัวนี้ก็เพื่อบรรยายคุณลักษณะของการเชื่อมต่ออุปกรณ์รับส่งข้อมูลปลายทาง (Data Terminal Equipment: DTE) กับอุปกรณ์สื่อสารข้อมูล (Data Communication Equipment: DCE) สำหรับผู้ใช้คอมพิวเตอร์ DTE หมายถึงตัวคอมพิวเตอร์ และ DCE หมายถึงโมเด็ม อุปกรณ์อื่นๆ เช่น เครื่องพิมพ์ที่รับสัญญาณแบบอนุกรมอาจจะเป็นได้ทั้ง DTE และ DCE ขึ้นอยู่กับผู้ผลิต

ความเร็วและระยะทางการเชื่อมต่อพอร์ตอนุกรม RS-232C สามารถเชื่อมต่อการถ่ายโอนข้อมูลได้จาก 0-20,000 บิตต่อวินาที ซึ่งเพียงพอสำหรับคอมพิวเตอร์ที่มีอัตราการถ่ายเทข้อมูล 110 ถึง 9,600 บิตต่อวินาที ความยาวของสายเชื่อมต่อสัญญาณตามมาตรฐานของพอร์ตอนุกรม RS-232C จำกัดแค่ 50 ฟุต ซึ่งเพียงพอสำหรับการสื่อสารคอมพิวเตอร์กับอุปกรณ์ภายนอก

2.9.1 องค์ประกอบของการรับส่งข้อมูลแบบอนุกรม

การสื่อสารแบบอนุกรมที่นิยมใช้กับเครื่องคอมพิวเตอร์นั้น เป็นการสื่อสารข้อมูลแบบอะซิงโครนัส (Asynchronous) นั่นคือ ต้องใช้สายสัญญาณเส้นเดียวทำหน้าที่ทั้งส่งข้อมูลและควบคุมการส่งข้อมูล ดังนั้นข้อมูลที่อ่านได้แต่ละบิตจากการส่งแบบอนุกรม ต้องถูกแยกแยะว่าใช้สำหรับจุดประสงค์ใด ซึ่งสามารถแบ่งได้เป็น 4 ส่วนดังนี้

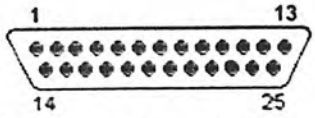
- บิตเริ่มต้น (Start Bit) ใส่ที่จุดเริ่มต้นเพื่อบอกฝ่ายรับข้อมูลว่าข้อมูลกำลังจะมาถึง
-บิตข้อมูล (Data Character) การส่งบิตข้อมูลจะส่งเป็นกลุ่ม มีขนาดโดยทั่วไปเป็น 7 หรือ 8 บิต
- บิตพาริตี้ (Parity Bit) ใช้ในการตรวจสอบความถูกต้องของข้อมูลที่ส่ง
- บิตจบ (Stop Bit) เป็นบิตที่ส่งมาปิดท้ายข้อมูล



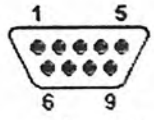
รูปที่ 2-22 การส่งข้อมูลอนุกรม

2.9.2 ลักษณะของคอนเน็กเตอร์แบบ D-Type

มาตรฐานการเชื่อมต่อแบบ RS-232C จะใช้คอนเน็กเตอร์แบบ DB-25 หรือ DB-9 ตัวผู้ ซึ่งคอนเน็กเตอร์แบบ DB-25 จะมีขาต่อใช้งานเพียง 9 เส้น เช่นเดียวกับคอนเน็คเตอร์แบบ DB-9 ซึ่งหัวต่อทั้ง 2 ชนิดนี้จะมีลักษณะการทำงานของสัญญาณเหมือนกัน แต่การจัดเรียงไม่เหมือนกัน โดยสามารถแสดงรูปร่างและตำแหน่งขาได้ดังรูปที่ 2-23 และ 2-24



รูปที่ 2-23 คอนเน็คเตอร์อนุกรม 25 ขา



รูปที่ 2-24 คอนเน็คเตอร์อนุกรม 9 ขา

DB-25	DB-9	ชื่อของสายสัญญาณ	ชนิดของสายสัญญาณ
8	1	Data Carrier Detect: DCD	Input
3	2	Received Data: RxD	Input
2	3	Transmitted Data: TxD	Output
20	4	Data Terminal Ready: DTR	Output
7	5	Signal Ground: GND	-
6	6	Data Set Ready: DSR	Input
4	7	Request To Send: RTS	Output
5	8	Clear To Send: CTS	Input
22	9	Ring Indicator: RI	Input

ตารางที่ 2.15 การจัดขาคอนเน็คเตอร์พอร์ตอนุกรมตามมาตรฐาน RS-232C แบบ DB-25 และ DB-9

สำหรับรายละเอียดหน้าที่การทำงานในแต่ละขาของพอร์ตอนุกรม RS-232C มีดังนี้

1. Data Carrier Detect: DCD หรืออาจเรียกว่า Carrier Detect: CD ขานี้จะแอคทีฟเมื่อมีการส่งสัญญาณพาห์จากอุปกรณ์สื่อสารข้อมูล เช่น โมเด็ม สำหรับการใช้งานปกติขานี้จะไม่ได้ถูกใช้งานมากนัก

2. Received Data: RD หรือ RxD ขานี้ใช้เพื่อรับสัญญาณอนุกรมเข้ามายังคอมพิวเตอร์ โดยนำข้อมูลที่อ่านได้เก็บไว้ในรีจิสเตอร์บัฟเฟอร์

3. Transmitted Data: TD หรือ TxD ขานี้ใช้เพื่อส่งข้อมูลออกจากคอมพิวเตอร์ โดยนำข้อมูลที่เก็บอยู่ในบัฟเฟอร์สำหรับส่งข้อมูลออกไป

4. Data Terminal Ready: DTR เป็นขาสัญญาณที่ส่งออกจากคอมพิวเตอร์เพื่อให้อุปกรณ์ปลายทางรับรู้ว่าต้องการติดต่อด้วย โดยขา DTR นี้จะต้องเชื่อมต่อกับขา DSR ของอุปกรณ์ปลายทาง

5. Signal Ground: GND ขากราวนด์ของระบบ

6. Data Set Ready: DSR ใช้คู่กับขา DTR เพื่อตรวจสอบการเชื่อมต่อระหว่างคอมพิวเตอร์กับอุปกรณ์ปลายทาง ซึ่งขา DSR นี้เป็นขาสำหรับรับข้อมูลจากภายนอกซึ่งถูกส่งมาจากขา DTR

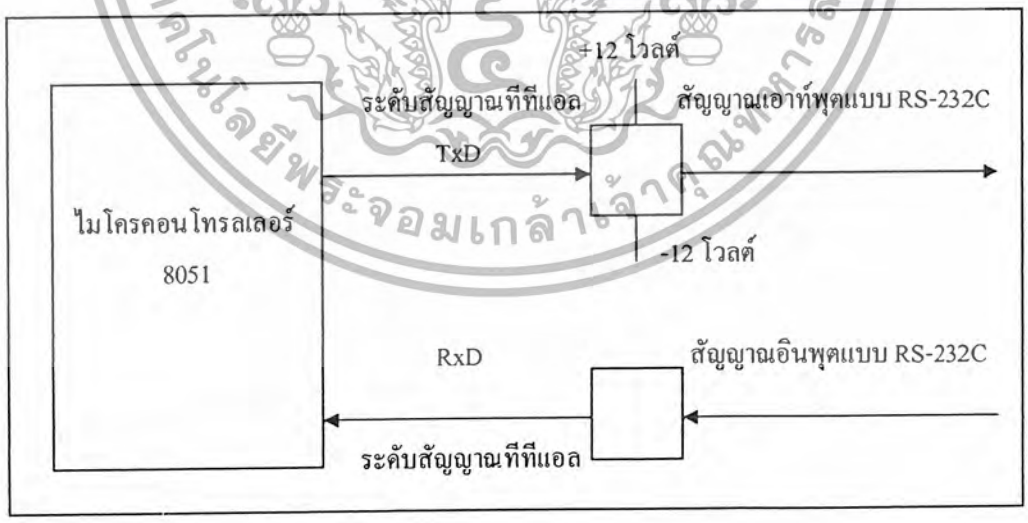
7. Request To Send: RTS เป็นขาสำหรับส่งสัญญาณร้องขอให้ทางอุปกรณ์ปลายทางส่งข้อมูลกลับมายังคอมพิวเตอร์ โดยขาที่รับสัญญาณ RTS ก็คือขา CTS

8. Clear To Send: CTS ขานี้จะคอยรับสัญญาณจากขา RTS เมื่อรับสัญญาณได้ ข้อมูลที่ขา TxD จะถูกส่งออกไป ขานี้จึงถูกใช้เพื่อตรวจสอบอุปกรณ์ต่อพ่วงว่าพร้อมที่จะรับข้อมูลหรือไม่

9. Ring Indicator: RI ใช้แสดงสถานะสัญญาณเรียกจากสายโทรศัพท์ ใช้เมื่อมีการเชื่อมต่อกับโมเด็ม

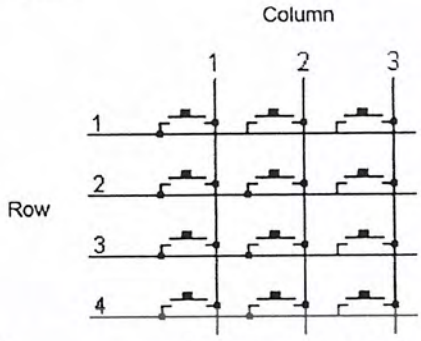
2.9.3 การเชื่อมต่อมาตรฐาน RS-232C

ในเชื่อมต่ออนุกรมเข้ากับอุปกรณ์คอมพิวเตอร์ต่างๆ มักจะกำหนดในการเชื่อมต่อตามมาตรฐาน RS-232C ทั้งนี้เพื่อให้มีการใช้งานเส้นสัญญาณหรือรูปแบบของตัวเชื่อมต่อที่สอดคล้องกัน จะได้ลดปัญหาการเข้ากันไม่ได้ระหว่างสัญญาณของอุปกรณ์ที่มาเชื่อมต่อทั้งสองด้าน เนื่องจากระดับแรงดันไฟฟ้าที่ใช้ และการแทนความหมายของระดับลอจิกตามมาตรฐานนี้แตกต่างไปจากที่ใช้งานกันในระบบดิจิตอลทั่วไป โดยระดับสัญญาณของพอร์ต RS-232C เป็นแบบสองขั้ว (Bipolar) ที่ระดับแรงดันไฟฟ้าทางด้านลบช่วง -3 โวลต์ ถึง -12 โวลต์ จะแทนค่าลอจิกสูง และแรงดันไฟฟ้าบวกช่วง +3 โวลต์ ถึง +12 โวลต์ แทนค่าลอจิกต่ำ จะเห็นได้ว่ามีความจำเป็นที่จะต้องเพิ่มอุปกรณ์หรือวงจรพิเศษเข้าไปเพื่อเปลี่ยนระดับแรงดันไฟฟ้าให้ในช่วง +3 โวลต์ ถึง +5 โวลต์ ซึ่งได้จากขาสัญญาณของตัวไมโครคอนโทรลเลอร์ 8051 ซึ่งเป็นระดับแรงดันไฟฟ้าที่สูงกว่าค่า +3 โวลต์ หรือต่ำกว่า -3 โวลต์ ดังรูปที่ 2-25



รูปที่ 2-25 การเปลี่ยนแปลงสัญญาณทีทีแอล (TTL)

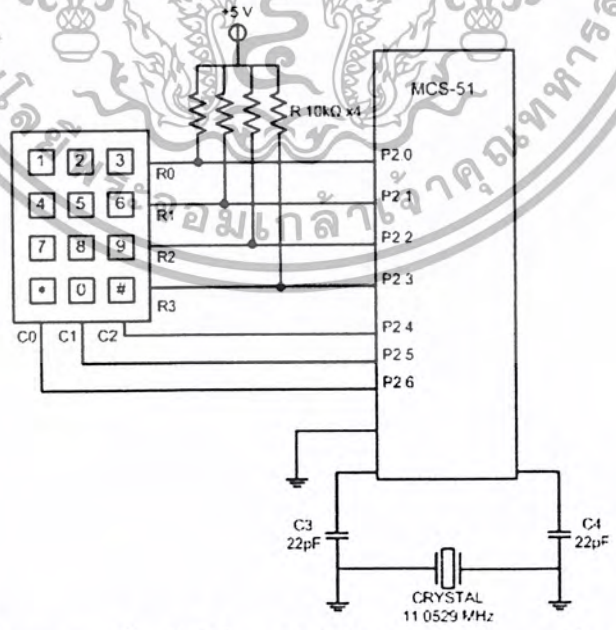
2.10 คีย์แพดหรือสวิตช์เมตริกซ์



รูปที่ 2-26 วงจรสวิตช์แบบเมตริกซ์หรือคีย์แพด

เป็นการพัฒนามาจากสวิตช์แบบธรรมดาที่ต่อเข้ากับไฟเลี้ยงหรือกราวด์โดยตรง ซึ่งมีข้อเสียเมื่อมีจำนวนของสวิตช์มากๆ ส่งผลให้ระบบหรือวงจรโดยรวมมีขนาดใหญ่และสิ้นเปลือง โดยการต่อวงจรแบบเมตริกซ์ (Matrix Switch) นี้จะประกอบไปด้วยสวิตช์ที่ต่อกันในแนวแกนตั้งที่เรียกว่าหลักหรือคอลัมน์ (column) และในแนวแกนนอนที่เรียกว่า แถวหรือโรว์ (row) ดังในรูปที่ 2-26 ถึงแม้ว่ากระบวนการจะมีความซับซ้อนมากกว่าแต่ก็สามารถรองรับการเพิ่มของสวิตช์ได้อย่างสะดวกเพียงทำการแก้ไขซอฟท์แวร์เท่านั้น ในการใช้งานทั่วไปเรียกเมตริกซ์สวิตช์นี้ว่า คีย์แพด (keypad)

2.10.1 การเชื่อมต่อคีย์แพดเข้ากับไมโครคอนโทรลเลอร์ MCS-51



รูปที่ 2-27 วงจรเชื่อมต่อคีย์แพดเข้ากับไมโครคอนโทรลเลอร์ MCS-51

ตัวอย่างวงจรแสดงในรูปที่ 2.-27 ใช้พอร์ต 2 ของไมโครคอนโทรลเลอร์เชื่อมต่อเข้ากับคีย์แพดทั้ง 7 เส้น คือสายคอลัมน์ 3 สาย C0-C2 และสายทางโรว์อีก 4 สาย คือ R0-R3 โดยที่ขาพอร์ต P2.0-P2.3 จะต้องต่อตัวต้านทานพูลอัปไว้เพื่อกำหนดสภาวะเริ่มต้นที่ไม่มีการกดคีย์ โดยไมโครคอนโทรลเลอร์จะทำการส่งข้อมูล "0" ไปยัง P2.6, P2.5 และ P2.4 ตามลำดับ ในทุกครั้งที่มีการส่งข้อมูลไปยังสายคอลัมน์ของคีย์แพด ไมโครคอนโทรลเลอร์จะทำการอ่านค่าที่ P2.0-P2.3 เข้ามาด้วย หากไม่มีการกด ค่าของ P2.0-P2.3 จะเป็น "1" ทั้งหมด แต่ถ้าหากมีการกดคีย์ ค่าของ P2.0-P2.3 ก็จะไม่เป็น 1111 เพื่อแจ้งให้ทราบว่ามีการกดคีย์แพดขึ้นแล้ว จากนั้นไมโครคอนโทรลเลอร์ก็จะทำการค้นหาตำแหน่งต่อไปและนำค่าตำแหน่งที่ได้ไปเปิดตารางข้อมูลเพื่อที่จะได้หมายเลขของคีย์ที่กดอย่างแท้จริง

2.11 โปรแกรมวิชวลเบสิก (Visual Basic)

ในปัจจุบัน ระบบปฏิบัติการ (Operating System) ในลักษณะของ Windows ได้เข้ามาแทนการปฏิบัติการในลักษณะเดิมซึ่งส่วนใหญ่ที่นิยมใช้กันอยู่คือ MS-DOS เนื่องจากรูปแบบของจอภาพที่ใช้ติดต่อกันระหว่างคอมพิวเตอร์ และผู้ใช้ อยู่ในรูปแบบของ Graphic User Interface (GUI) ที่ใช้รูปภาพแทนคำสั่งต่างๆทำให้ผู้ใช้สามารถสร้างโปรแกรมได้ง่ายขึ้น ซึ่งต่างจาก MS-DOS ที่รูปแบบของคำสั่งจะอยู่ในรูปแบบของตัวอักษร และเป็นการป้อนทีละบรรทัด หรือเรียกว่า "Command Line" ซึ่งผู้ใช้จะต้องเรียนรู้ และจดจำรูปแบบของแต่ละคำสั่งให้ถูกต้องและแม่นยำ จึงจะใช้งานโปรแกรมนั้นๆได้เป็นอย่างดี และด้วยเหตุนี้ จึงทำให้ โปรแกรมเมอร์ซึ่งแต่เดิมพัฒนาโปรแกรมอยู่บน MS-DOS ทันมาพัฒนาโปรแกรมบน Windows แทน

วิชวลเบสิก(Visual Basic) เป็นภาษาคอมพิวเตอร์ที่นิยมนำมาใช้พัฒนาโปรแกรมบน Windows เนื่องจากเป็นภาษาคอมพิวเตอร์ที่ใช้เทคโนโลยีในลักษณะ Visualize ซึ่งเพียงแต่เลือก Control ที่เหมาะสม แล้ววางลงบน Form ก็สามารถที่จะสร้างจอภาพที่ใช้สำหรับติดต่อกันผู้ใช้ รวมทั้งการใช้เทคนิคการเขียนโปรแกรมรวมแบบ Event-driven ซึ่งเป็นการเขียนโปรแกรมเพื่อกำหนดขั้นตอนการทำงานให้กับ Control ต่างๆ ที่สร้างขึ้นตามเหตุการณ์ต่างๆที่เกิดขึ้น เช่น การเลื่อนเม้าส์ หรือการรับข้อมูลจากคีย์บอร์ด ฯลฯ เป็นต้น

2.11.1 โปรแกรมติดต่อและควบคุมผ่านพอร์ตอนุกรม

คอนโทรลที่ทำให้วิชวลเบสิกสามารถติดต่อผ่านพอร์ตอนุกรมได้คือคอนโทรล MsComm โดยมีพร็อพเพอร์ตี้ที่สำคัญ คือ

1. CommPort ใช้ในการกำหนดหมายเลขของพอร์ตอนุกรมที่เราต้องการจะติดต่อ โดยมีรูปแบบการใช้งานดังนี้

   Object.Commport = value

   ยกตัวอย่างเช่น ถ้าเขียนโปรแกรมติดต่อกับพอร์ต Com1 จะเขียนเป็น

   MSComm1.CommPort = 1

2. Settings ใช้กำหนดอัตราบอด (Baud Rate) หรือความเร็วในการส่งข้อมูลมีหน่วยเป็นบิตต่อวินาที, พาริตี้, จำนวนของบิตข้อมูล, จำนวนของบิตปิดท้าย โดยมีรูปแบบการใช้งานดังนี้

   Object.Settings = value

   ยกตัวอย่างเช่น ถ้าเขียนโปรแกรมใช้งานที่อัตราบอดเท่ากับ 9,600 บิตต่อวินาที, ไม่มีพาริตี้, จำนวนบิตข้อมูลเท่ากับ 8 บิต และมีบิตปิดท้าย 1 บิต จะเขียนได้เป็น

   MSComm1.Settings = "9600, N, 8,1"

3. PortOpen ใช้สำหรับเปิดและปิดการใช้งานพอร์ตอนุกรม ถ้าจะเปิดใช้งานพอร์ตอนุกรมให้กำหนดค่า value เป็น True ถ้าจะปิดพอร์ตอนุกรมให้กำหนดค่า value เป็น False โดยมีรูปแบบการใช้งานดังนี้

   Object.PortOpen = value

4. InBuffersize เป็นการกำหนดขนาดของ Buffer ในการรับข้อมูลเข้ามา โดยมีรูปแบบการทำงานดังนี้

   Object.InBuffersize = value

5. OutBuffersize เป็นการกำหนดขนาดของ Buffer ในการส่งข้อมูลออกไป โดยมีรูปแบบการทำงานดังนี้ Object.OutBuffersize = value

6. Inputlen เป็นการกำหนดค่าของข้อมูลที่อ่านจาก Buffer ภาครับ โดยมีรูปแบบการทำงานดังนี้

   Object.Inputlen = value

7. Input ใช้ในการอ่านข้อมูลจากพอร์ตอนุกรม โดยมีรูปแบบการอ่านดังนี้

   value = Object.Input

8. Output ใช้ในการส่งข้อมูลออกไปจากพอร์ตอนุกรม โดยมีรูปแบบการส่งดังนี้

   Object.Output = value

2.12 โปรแกรมจาวา (Java)

2.12.1 พื้นฐานของโปรแกรมส่งข้อมูลผ่านระบบเครือข่าย

ในการเขียนโปรแกรมรับส่งข้อมูลต่างๆจากเครื่องคอมพิวเตอร์หนึ่ง ไปสู่อีกเครื่องหนึ่งในระบบเครือข่ายนั้น เป็นเรื่องที่ค่อนข้างยุ่งยาก และจำเป็นต้องใช้ผู้เชี่ยวชาญในการสร้างโปรแกรมดังกล่าว ซึ่งผู้เขียนโปรแกรมนั้นต้องสร้างโปรแกรมทั้งด้านผู้รับและผู้ส่ง เนื่องจากทางด้านผู้ส่งจะต้องนำข้อมูลมาตัดแบ่งออกเป็นส่วนย่อยๆเรียกว่า Packet และในแต่ละ Packet จะมีส่วนประกอบของ header และ payload ผู้เขียนโปรแกรมจึงจำเป็นต้องทราบวิธีการสร้าง Packet และส่ง Packet ผ่านระดับชั้นของโปรแกรมหลายชั้น เพื่อทำการเพิ่มเติมและปรับเปลี่ยนข้อมูลให้เหมาะสมในการเดินทางผ่านตัวกลางไปสู่ผู้รับได้ และทางด้านผู้รับต้องทราบวิธีการรับ Packet จากระบบเครือข่ายขึ้นมาประกอบเป็นลำดับที่ถูกต้อง เพื่อทำการเปลี่ยนคืนกลับเป็น Packet ดังเดิม

ในโปรแกรมจาวามี Socket ซึ่งใช้ในการเขียนโปรแกรมส่งข้อมูลผ่านระบบเครือข่าย โดยจะซ่อนรายละเอียดโครงสร้างการทำงานของ packet ข้อมูลต่างๆ ผู้เขียนโปรแกรมจึงสามารถเขียนโปรแกรมรับส่งข้อมูลผ่านระบบเครือข่ายได้เหมือนกับการอ่านเขียนข้อมูลจาก stream ซึ่งนอกจากจะไม่ขึ้นกับแพลตฟอร์มแล้ว ผู้เขียนโปรแกรมยังไม่จำเป็นต้องเข้าใจรายละเอียดเกี่ยวกับ protocol layer, hardware หรือตัวกลางที่ใช้ในการสื่อสาร ดังนั้น เมื่อมี socket แล้ว การเขียนโปรแกรมรับส่งข้อมูลผ่านระบบเครือข่ายจึงง่ายขึ้นอย่างมาก

เมื่อเราสร้าง Socket จะทำการติดต่อไปยังเครื่องเป้าหมายทันทีโดยใช้ TCP/IP หลังจากนั้นเครื่องเริ่มต้นจะสามารถขอ input stream และ output stream จาก socket เพื่อทำการอ่านเขียนข้อมูลไปที่เครื่องเป้าหมาย ดังนั้นจึงจำเป็นต้องมี protocol ในการสื่อสารข้อมูลระหว่างเครื่องทั้งสอง โดยเราจะกำหนดให้แต่ละพอร์ตต้องมี protocol ที่แน่นอน เพื่อให้เราสามารถติดต่อไปยังพอร์ตนั้นได้ ค่าของพอร์ตนั้นต้องเป็นเลขจำนวนเต็ม โดยมีค่าได้ตั้งแต่ 1 ถึง 65,535 แต่ในเครื่องทั่วไปจะถูกกำหนดพอร์ตมาตรฐานไว้ตั้งแต่ 1 ถึง 1024 เช่น เบอร์ 21 เป็น ftp, 23 เป็น telnet, 25 เป็น smtp, 80 เป็น http เป็นต้น ดังนั้นหากเราจะเปิดพอร์ตเพื่อใช้ในการติดต่อกันเอง ควรเลือกค่าที่สูงกว่า 1024 ขึ้นไป และควรตรวจสอบด้วยว่ามีโปรแกรมอื่นที่กำลังใช้งานพอร์ตเบอร์นั้นอยู่ก่อนแล้วหรือไม่

โดยปกติแล้ว ในการส่งข้อมูลระหว่างเครื่องสองเครื่องผ่านทาง Socket จำเป็นต้องมีเครื่องด้านหนึ่งอุทิศตัวเองเป็นผู้รอรับการติดต่อ ซึ่งเราจะเรียกเครื่องที่ทำหน้าที่รอรับการติดต่อนี้ว่าเซิร์ฟเวอร์ (server) และเครื่องที่เป็นผู้ติดต่อเข้ามาเรียกว่าไคล์แอนท์ (client) โดยที่ภาษาจาวานั้น จะมีคลาส ServerSocket ที่มีกลไกในการรอรับการเชื่อมต่อจากเครื่องไคล์แอนท์ที่พอร์ตเบอร์หนึ่ง และเมื่อมีการเชื่อมต่อเข้ามาแล้ว ก็จะทำการเชื่อมการติดต่อนั้น และสร้างออกมาเป็น socket ซึ่งต้องใช้

หนึ่ง socket ในการติดต่อกับหนึ่งไคล์แอนท์ เพื่อให้เซิร์ฟเวอร์สามารถทำการรับส่งข้อมูลกับไคล์แอนท์ได้ต่อไป

2.12.2 ส่วนของโปรแกรมการเข้ารหัสลับและถอดรหัสลับ

ความเข้าใจพื้นฐานในเรื่องของการเข้ารหัสลับและถอดรหัสลับ

การเข้ารหัสลับข้อมูล โดยพื้นฐานแล้วจะเกี่ยวข้องกับวิธีการทางคณิตศาสตร์เพื่อใช้ในการป้องกันข้อมูลหรือข้อความตั้งต้นที่ต้องการส่งไปถึงผู้รับ โดยที่การเข้ารหัสลับนั้น เป็นการนำข้อมูลประเภท Clear text มาเข้าสู่กระบวนการบางอย่างกับส่วนของข้อมูลสตริงชุดหนึ่ง ที่มีขนาดไม่ใหญ่มาก หรือที่เรียกว่ากุญแจรหัสนั้น ซึ่งกระบวนการดังกล่าว ส่งผลให้บุคคลที่สามที่ต้องการเข้าถึงข้อมูลที่เป็นความลับ หรือบุคคลที่ต้องการจะโจมตีนั้น ไม่สามารถล่วงรู้ถึงข้อมูลที่ถูกเข้ารหัสลับหรือ cipher text ได้ หากปราศจากกุญแจรหัสที่ถูกต้อง ส่วนการถอดรหัสลับนั้นเป็นกระบวนการย้อนกลับของการเข้ารหัสลับ นั่นคือการนำข้อมูลประเภท cipher text มาเข้าสู่กระบวนการการถอดรหัสลับ โดยใช้กุญแจรหัส ซึ่งจะได้ผลออกมาเป็นข้อมูลประเภท Clear text ซึ่งเป็นข้อมูลก่อนที่จะเข้าสู่กระบวนการเข้ารหัสลับนั่นเอง

SecretKeyFactory Class

เป็นการทำงานในส่วนของการจัดการค่าของกุญแจรหัส ซึ่งถูกจัดเก็บอยู่ใน Factory โดยจะทำหน้าที่ในการแปลงค่ากุญแจรหัส ให้เป็นค่าของกุญแจรหัสเฉพาะซึ่งมีรูปแบบที่เหมาะสมต่อการใช้งาน เนื่องจากในส่วนของการใช้งานนั้น จะขึ้นอยู่กับในแต่ละผู้ให้บริการ โดยอาจจะมีพื้นฐานอยู่ทั้งบนฮาร์ดแวร์ และซอร์ฟแวร์ของผู้ให้บริการในแต่ละรายอีกด้วย ซึ่งคลาสนี้ จะสามารถบริหารการใช้งานค่าของกุญแจรหัส ให้เหมาะสมกับผู้ให้บริการในแต่ละราย ดังตัวอย่างโปรแกรมด้านล่าง

```
byte[] desKeyData = { (byte)0x01, (byte)0x02, (byte)0x03,
(byte)0x04, (byte)0x05, (byte)0x06, (byte)0x07, (byte)0x08 };
DESKeySpec desKeySpec = new DESKeySpec(desKeyData);
SecretKeyFactory keyFactory =
SecretKeyFactory.getInstance("DES");
SecretKey secretKey = keyFactory.generateSecret(desKeySpec);
```

แนวทางหนึ่งในการจัดการแปลงค่าของกุญแจรหัสโดยไม่ขึ้นอยู่กับแนวคิดของผู้ให้บริการในแต่ละรายนั้น เราจะใช้คลาสของ `javax.crypto.spec.SecretKeySpec` โดยจะใช้อินเทอร์เฟซของ `javax.crypto.SecretKey` ดังตัวอย่าง

```
byte[] desKeyData = { (byte)0x01, (byte)0x02, ...};
SecretKeySpec secretKey = new SecretKeySpec(desKeyData, "DES");
```

การสร้าง Cipher

ในขั้นตอนนี้จะเป็นการสร้าง Instance ของ cipher โดยที่เราสามารถ get instance ของ cipher ในแต่ละชนิดได้ ซึ่งเราจำเป็นต้องระบุชื่อของ cipher เพื่อร้องขออัลกอริทึมที่เราต้องการ และอาจจะระบุโหมดของ cipher และเทคนิคในการ padding ต่างๆเพิ่มได้อีกด้วย โดยสองอย่างหลังนั้นเป็นเพียง optional ตัวอย่างเช่นโค้ดโปรแกรมด้านล่างนี้ เป็นการเข้ารหัสลับ และการถอดรหัสลับโดยใช้อัลกอริทึม DES และใช้โหมด Electronic Codebook mode (ECB) และใช้เทคนิค PKCS #5 ในการ padding

```
Cipher desCipher;
desCipher = Cipher.getInstance("DES/ECB/PKCS5Padding");
desCipher.init(Cipher.ENCRYPT_MODE, desKey);
byte[] cleartext = "This is just an example".getBytes();
byte[] ciphertext = desCipher.doFinal(cleartext);
desCipher.init(Cipher.DECRYPT_MODE, desKey);
byte[] cleartext1 = desCipher.doFinal(ciphertext);
```

ซึ่งเราจะได้ค่าของ `cleartext` และ `cleartext1` ที่มีค่าเท่ากัน

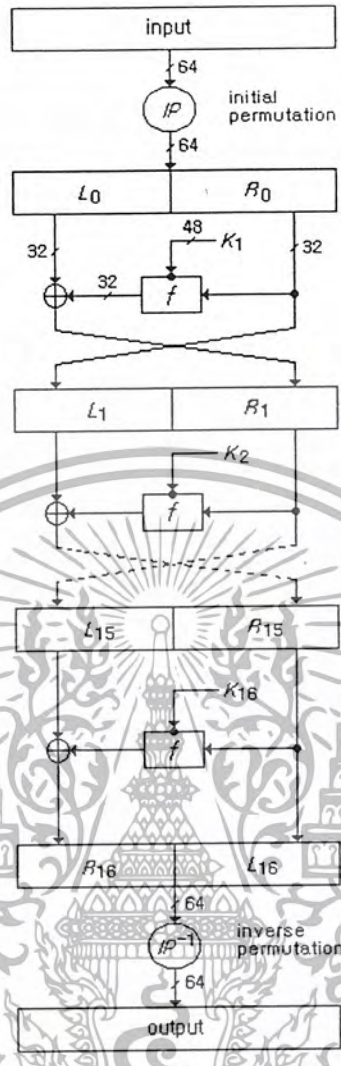
หลักการทำงานพื้นฐานของอัลกอริทึม Data Encryption Standard (DES)

DES เป็นอัลกอริทึมที่พัฒนาขึ้นโดย IBM Corporation โดยได้รับการอนุมัติจากรัฐบาลสหรัฐฯ ในปีค.ศ.1977 โดยในระหว่างการพัฒนานั้น ได้มีการปรับแก้หลายจุดตามคำวิจารณ์จาก National Security Agency (NSA) ก่อนจะกลายมาเป็นมาตรฐานที่ใช้กันจนปัจจุบัน ซึ่ง DES นั้นเป็นอัลกอริทึมที่ถูกนักถอดรหัสลับหลายๆท่าน ทำการศึกษากันอย่างจริงจังร่วม 20 ปี แต่ตลอดระยะเวลาที่ผ่านมา พบว่ายังไม่มีวิธีการใดเลยที่จะสามารถเจาะผ่านการเข้ารหัสลับแบบนี้ได้ ยกเว้นวิธีการ Brute-force Method เพียงวิธีเดียวเท่านั้น

อัลกอริทึม DES นั้น จะประกอบด้วยส่วนสำคัญสองส่วนคือ ขั้นตอนการสลับสับเปลี่ยนบิตข้อมูลตามแบบแผน และขั้นตอนการกระจายบิตข้อมูลด้วยกุญแจรหัส ซึ่งกุญแจรหัสที่ใช้มีขนาด 8 ไบต์ หรือ 64 บิต แต่จะใช้งานจริงเพียง 7 ไบต์ หรือ 56 บิตเท่านั้น คือจะแบ่งออกเป็นบิตของกุญแจรหัสขนาด 56 บิต กับ parity bit ขนาด 8 บิต ซึ่งหมายความว่ากุญแจรหัสที่สามารถใช้ในการเข้ารหัสแบบนี้จะสามารถมีได้มากถึง 2^{56} หรือ $7x10^{16}$ กุญแจรหัสเลยทีเดียว ซึ่งในการเข้ารหัสลับนั้นจะแบ่งเป็นบล็อกขนาด 64 บิต และแบ่งออกเป็นสองซีกซ้าย-ขวา แล้วทำการเข้ารหัสลับด้วยกุญแจรหัส จากนั้น จะทำการสลับด้านซ้าย-ขวา แล้วสร้างกุญแจรหัสใหม่จากกุญแจรหัสเดิมมาเข้ารหัสลับอีกครั้งหนึ่ง ทำแบบนี้ 16 รอบ จึงจะได้ข้อมูลออกมาเป็น cipher text (ดังรูปที่ 2-28) ซึ่งจากขนาดของกุญแจรหัสดังกล่าวนั้น ไม่ใช่เรื่องง่ายเลยหากเราต้องการจะค้นหากุญแจรหัสที่ถูกต้องในการ

ถอดรหัสลับแบบนี้ ซึ่งหากเราลองใช้เครื่องคอมพิวเตอร์รุ่นเก่าๆ เช่นหน่วยประมวลผล 80486 ซึ่งจะใช้เวลา 30 วินาที ต่อการถอดรหัสข้อมูลหนึ่งชุด ทำให้ต้องใช้เวลาในการสุ่มหากุญแจรหัสจนพบมากถึง 17.5 ล้านปี เลยทีเดียว

จากการทำงานข้างต้น ถือว่าอัลกอริทึม DES มีความน่าเชื่อถือสูงในระดับหนึ่ง แต่อย่างไรก็ตาม เนื่องจากการพัฒนาเทคโนโลยีคอมพิวเตอร์ในปัจจุบันนั้น สามารถพัฒนาได้อย่างรวดเร็วมาก ซึ่งจากการสำรวจ พบว่าหากเราใช้หน่วยประมวลผล Pentium ความเร็วสัญญาณนาฬิกา 200 MHz จะใช้เวลาในการสุ่มหากุญแจรหัสจนพบภายใน 2.33 ล้านปี ซึ่งหากต้องการถอดรหัสลับข้อมูลจริงๆ แล้ว อาจไม่จำเป็นต้องทำจากเพียงเครื่องเดียว ซึ่งหากใช้เครื่องคอมพิวเตอร์ที่มีความเร็วดังกล่าว จำนวน 1,000,000 เครื่อง ช่วยกันถอดรหัสลับจะใช้เวลาเพียง 2.33 ปี หรือยิ่งไปกว่านั้นหากเป็นระดับของเครื่องซุปเปอร์คอมพิวเตอร์ด้วยแล้ว จะสามารถใช้เวลาในการถอดรหัสลับข้อมูลได้เพียงไม่กี่ชั่วโมงเท่านั้น ซึ่งถึงแม้ว่าผู้โจมตีอาจไม่มีความพยายามถึงขนาดนั้น แต่ก็ถือว่า DES ได้ถูกทำลายความน่าเชื่อถือลงไปแล้ว ทำให้เกิดวิธีที่ประยุกต์การเข้ารหัสลับที่สลับซับซ้อนมากยิ่งขึ้น ด้วยการเข้ารหัสลับข้อมูลด้วยกุญแจรหัส 2-3 ชุดตามลำดับ ซึ่งรู้จักกันในชื่อ Triple DES โดยสามารถเพิ่มความน่าเชื่อถือได้ในระดับหนึ่ง แต่ก็ทำให้การเข้ารหัสลับและถอดรหัสลับใช้เวลานานยิ่งขึ้น และยังสามารถถูกโจมตีด้วยวิธีการสุ่มกุญแจรหัสได้เช่นเดิม อย่างไรก็ตาม ทั้ง DES และ Triple DES ก็มีการประยุกต์การเข้ารหัสลับต่อไปอีกหลายรูปแบบอย่างต่อเนื่อง ซึ่งทำให้การเข้ารหัสลับข้อมูลด้วยวิธี DES ยังคงได้รับความน่าเชื่อถือต่อไปอีกระยะหนึ่ง

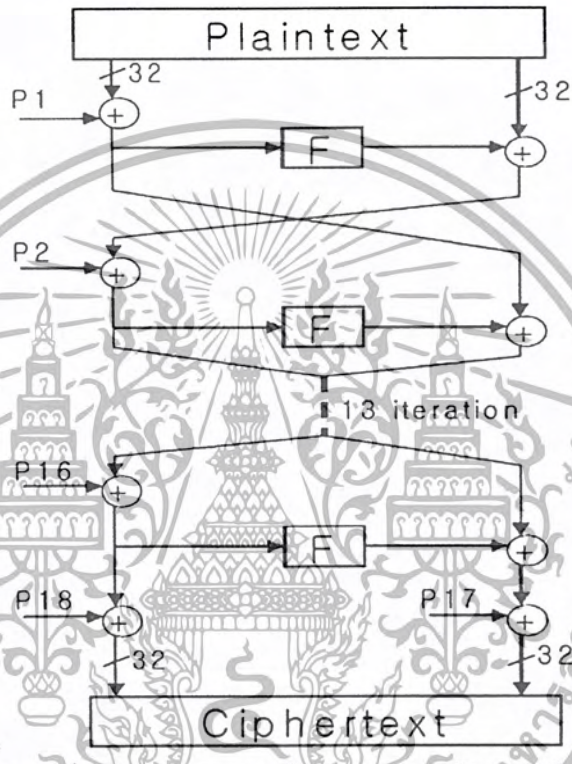


รูปที่ 2-28 แสดงบล็อกไดอะแกรมการเข้ารหัสลับโดยใช้อัลกอริทึม DES

หลักการทำงานพื้นฐานของอัลกอริทึม Blowfish

อัลกอริทึม Blowfish นั้น ได้รับการพัฒนาโดย Bruce Schneier โดยมีจุดมุ่งหมายในการแทนที่อัลกอริทึม DES และ IDEA ซึ่ง Blowfish นั้นมีข้อได้เปรียบอยู่หลายประการคือ เป็นอัลกอริทึมที่มีความรวดเร็วในการทำงาน มีขนาดเล็กกระทัดรัด และใช้การเข้ารหัสแบบบล็อค ซึ่งกุญแจรหัสที่ใช้งานนั้น เป็นกุญแจรหัสที่สามารถเปลี่ยนแปลงขนาดได้ (variable-length key) โดยเริ่มจากกุญแจรหัสที่มีขนาดเล็กๆไปจนถึงขนาด 448 บิต ซึ่งทำให้เกิดความยืดหยุ่นสูงในการเลือกใช้กุญแจรหัส และสามารถนำไปประยุกต์ใช้งานกับฮาร์ดแวร์ รวมทั้งอัลกอริทึมยังได้รับการออกแบบมาให้ทำงานอย่างเหมาะสมกับหน่วยประมวลผลทั้งขนาด 32 และ 64 บิตอีกด้วย นอกจากนี้ อัลกอริทึม Blowfish นั้นได้เปิดเผยสู่สาธารณะ และไม่ได้มีการจดสิทธิบัตรใดๆอีกด้วย

อัลกอริทึม Blowfish นั้น สามารถแบ่งการทำงานออกได้เป็น 2 ส่วน คือ ส่วนของการขยายค่าของกุญแจรหัส ซึ่งจะทำการแปลงค่าของกุญแจรหัส ซึ่งมีขนาดได้มากที่สุด 448 บิต ไปเป็นค่าอาเรย์ของกุญแจรหัสย่อย (p-array) ซึ่งมีขนาดทั้งหมด 4,168 ไบต์ และส่วนที่ 2 คือส่วนของการเข้ารหัสลับ ซึ่งกระบวนการเข้ารหัสลับจะทำ 16 รอบ โดยจะใช้เพียง 1 กุญแจรหัสย่อยเท่านั้นในการทำ XOR แต่ละรอบ ดังรูปที่ 2-29



รูปที่ 2-29 แสดงบล็อกไดอะแกรมการเข้ารหัสลับโดยใช้อัลกอริทึม Blowfish

บทที่ 3

การออกแบบ

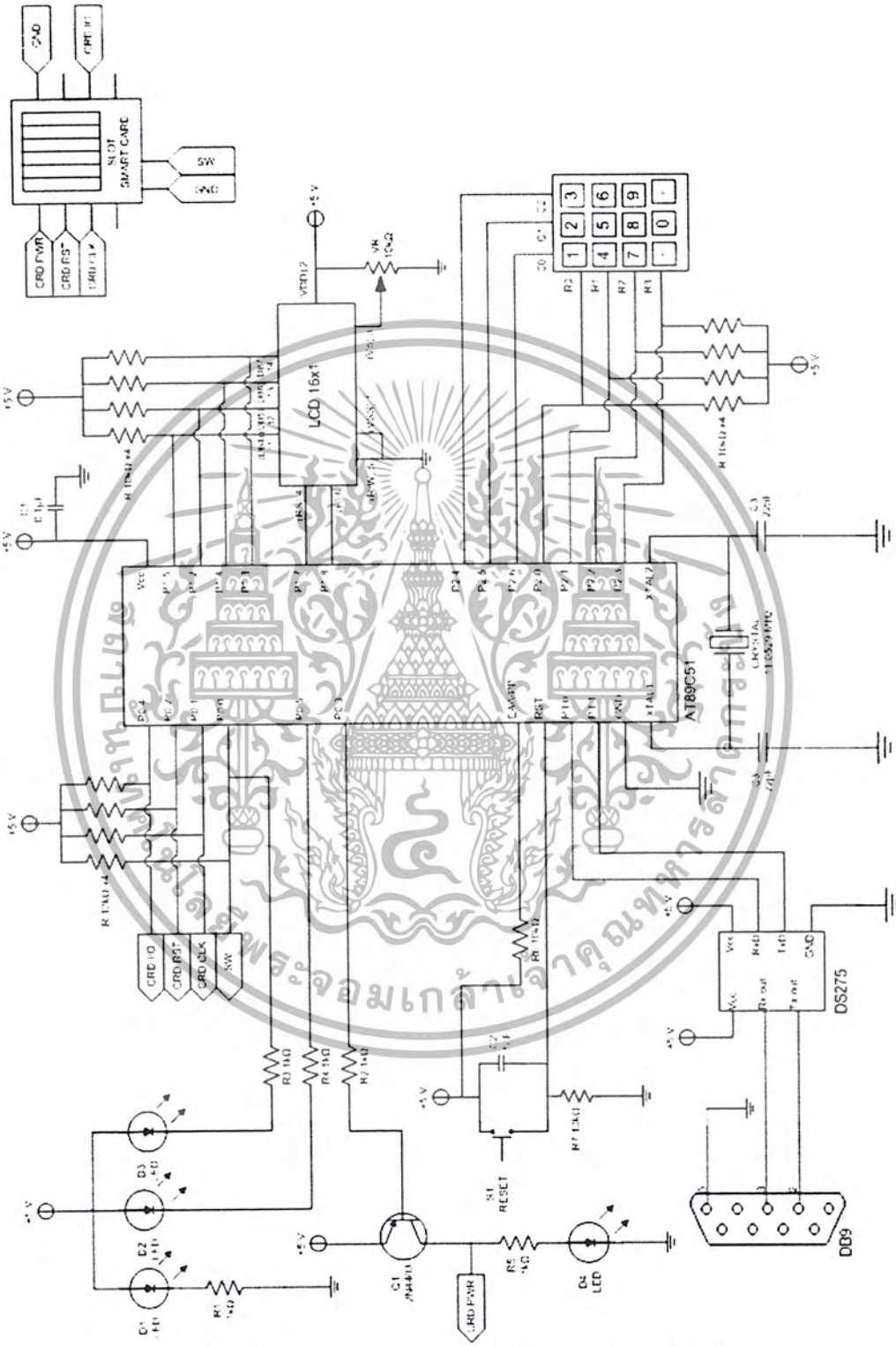
ในโครงงานการส่งข้อมูลที่เข้ารหัสลับโดยสมาร์ตการ์ดนี้ จะมีส่วนประกอบหลักๆที่สำคัญ คือ เครื่องอ่าน-เขียนสมาร์ตการ์ด ซึ่งควบคุมการทำงานโดยไมโครคอนโทรลเลอร์ ติดต่อกับพอร์ตอนุกรมของเครื่องคอมพิวเตอร์ และในส่วนของโปรแกรมการรับ-ส่งข้อมูลแบบเข้ารหัสลับผ่านทางระบบเครือข่าย โดยจะมีทางด้านฝั่งเซิร์ฟเวอร์หรือผู้ดูแลระบบ ซึ่งทำหน้าที่เขียนข้อมูลกุญแจรหัสลงในบัตรสมาร์ตการ์ด เพื่อแจกจ่ายให้กับผู้ที่ต้องการใช้งาน และมีการเก็บข้อมูลการใช้งานและสถานะของผู้ใช้งานต่างๆอยู่ใน Database อีกด้วย ซึ่งแสดงในรูปที่ 3-1

3.1 ภาพรวมของระบบ

รูปที่ 3-1 แสดงภาพรวมของระบบ

รายละเอียดการทำงานของระบบการส่งข้อมูลที่เข้ารหัสลับโดยสมาร์ตการ์ดนั้น สามารถแบ่งการทำงานออกเป็นส่วนต่างๆคือ เครื่องอ่าน-เขียนสมาร์ตการ์ดนั้นถูกควบคุมการทำงานด้วยไมโครคอนโทรลเลอร์ AT89C51 ซึ่งมี LED และจอแสดงผล LCD แสดงสถานะต่างๆ และจะมีส่วนของโปรแกรมที่ใช้ควบคุมการทำงานไมโครคอนโทรลเลอร์ ซึ่งเป็นโปรแกรมที่ใช้ในการอ่าน-เขียนข้อมูลหรือกุญแจรหัสลงในบัตรสมาร์ตการ์ดได้ และโปรแกรมที่ใช้ส่งข้อมูลแบบเข้ารหัสลับ โดยทางฝ่ายผู้ส่ง จะนำข้อมูลที่ต้องการจะส่งมาทำการเข้ารหัสลับกับค่ากุญแจรหัสที่ถูกเขียนลงในบัตรสมาร์ตการ์ดโดยผู้ดูแลระบบ และทางฝ่ายผู้รับจะมีโปรแกรมในลักษณะเดียวกันนี้ นำข้อมูลที่ได้รับมาทำการถอดรหัสลับ โดยใช้ค่ากุญแจรหัสที่อยู่ในสมาร์ตการ์ดเช่นเดียวกัน

3.2 การออกแบบทางด้านฮาร์ดแวร์



รูปที่ 3-2 แสดงวงจรภายในของเครื่องอ่าน-เขียนสมาร์ตการ์ด

จากรูปที่ 3-2 แสดงวงจรของเครื่องอ่าน-เขียนสมาร์ตการ์ด จะเห็นว่ามีช่องเสียบบัตรสมาร์ตการ์ด (Slot Smart card) ซึ่งถูกต่อเข้ากับขาของไมโครคอนโทรเลอร์ โดยที่ขา 35 ต่อเข้ากับตำแหน่ง CRD IO ของช่องเสียบบัตรสมาร์ตการ์ด ซึ่งเป็นขาที่ใช้เขียนหรืออ่านข้อมูลจากบัตร ที่ขานี้จะมี LED2 เป็นตัวแสดงผลเมื่อไมโครคอนโทรลเลอร์เขียนหรืออ่านข้อมูลจากบัตร โดย LED2 จะดิดหรือดับตามสถานะของลอจิกที่ขา 35 ของไมโครคอนโทรลเลอร์ ขา 37 ต่อเข้ากับตำแหน่ง CRD RST ของช่องเสียบบัตรสมาร์ตการ์ด ซึ่งเป็นขาที่ใช้รีเซตการทำงานของบัตร ขา 38 ต่อเข้ากับตำแหน่ง CRD CLK ของช่องเสียบบัตรสมาร์ตการ์ด ซึ่งขานี้เป็นขาที่ใช้ป้อนสัญญาณนาฬิกาอ้างอิงให้บัตรสมาร์ตการ์ด ขา 39 ต่อเข้ากับตำแหน่ง SW ของช่องเสียบบัตรสมาร์ตการ์ด ซึ่งเป็นขาที่ใช้ตรวจสอบและแสดงผลว่ามีการเสียบบัตรเข้ามายังช่องเสียบบัตรแล้วหรือยัง ที่ขานี้จะมี LED3 เป็นตัวแสดงผล ซึ่งหลอดจะสว่างเมื่อบัตรถูกเสียบเข้ามายังช่องเสียบบัตร และขา 36 ต่อกับตำแหน่ง CRD PWR ของช่องเสียบบัตรสมาร์ตการ์ด ซึ่งเป็นขาควบคุมการจ่ายไฟเลี้ยงให้กับบัตรด้วยการส่งให้ Q1 เปิดหรือปิดวงจร ที่ขานี้จะมี LED4 เป็นตัวแสดงผล เมื่อไฟเลี้ยงถูกจ่ายไปยังบัตร LED4 ก็จะสว่าง และจะมี LED1 คอยแสดงผลว่าวงจรกำลังทำงานอยู่

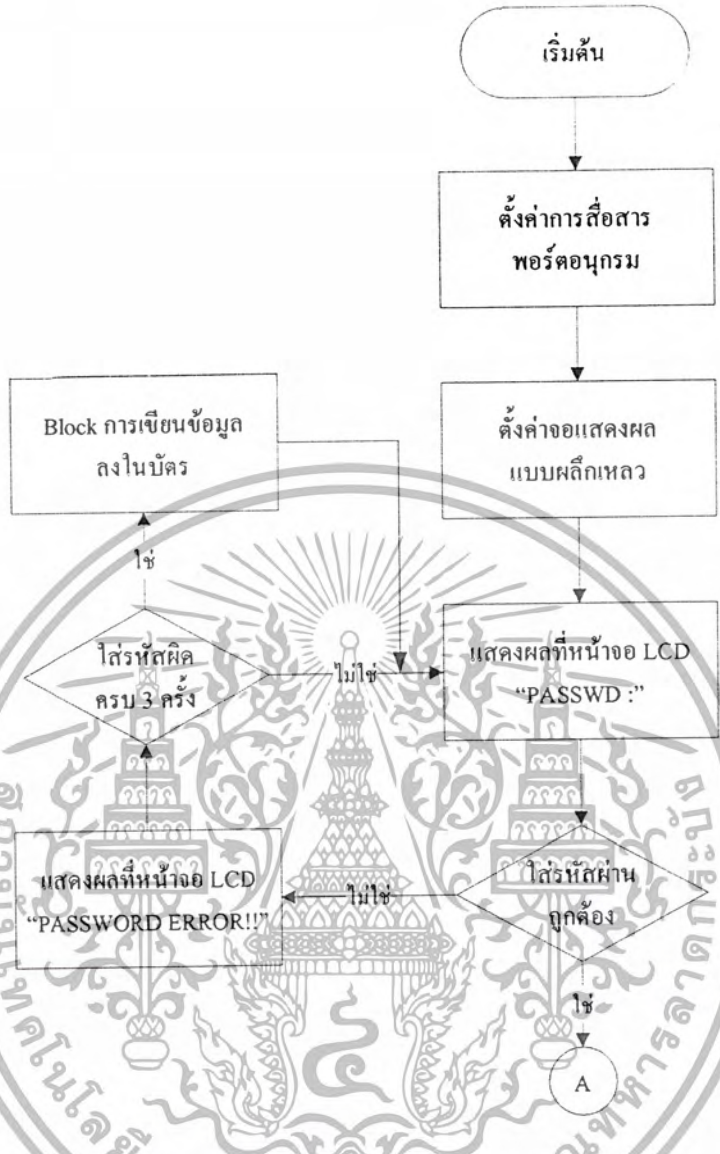
ที่ขา 10 ของไมโครคอนโทรลเลอร์เป็นขาสำหรับรับข้อมูลเข้าหรือ RxD และที่ขา 11 เป็นขาสำหรับส่งข้อมูลออกหรือ TxD ทั้งสองขาจะทำการเชื่อมต่อเข้ากับวงจรสื่อสารกับพอร์ตอนุกรมเพื่อสื่อสารข้อมูลกับโปรแกรมควบคุมคอมพิวเตอร์ ซึ่งมีไอซีเบอร์ DS275 เป็นหัวใจการทำงานหลักทำหน้าที่เป็นตัวกลางในการติดต่อ โดยไมโครคอนโทรลเลอร์อ่านข้อมูลที่รับเข้ามาหรือส่งออกไปที่รีจีสเตอร์บัฟเฟอร์ ที่ขา 3-6 ซึ่งเป็นพอร์ต 1 (P1) ของไมโครคอนโทรเลอร์ต่อเข้ากับขาข้อมูลทั้ง 4 เส้นของจอแสดงผลแบบผลึกเหลวขนาด 16 ตัวอักษร 1 บรรทัด ต่อมาขา 8 ซึ่งเป็น P1.7 ของไมโครคอนโทรลเลอร์จะต่อเข้ากับขา RS หรือขาที่ 4 ของจอแสดงผลแบบผลึกเหลว ส่วนขา 7 ซึ่งเป็น P1.6 ของไมโครคอนโทรลเลอร์จะต่อเข้ากับขา E หรือขาที่ 6 ของจอแสดงผลแบบผลึกเหลว และที่ขา 3 ของจอแสดงผลแบบผลึกเหลวจะต่อกับตัวต้านทานปรับค่าได้ (VR) ขนาด 10 kΩ ซึ่งจะเป็นตัวปรับความเข้มของจอแสดงผลแบบผลึกเหลว สุดท้ายในส่วนของคีย์แพด จะใช้พอร์ต 2 ของไมโครคอนโทรลเลอร์เชื่อมต่อเข้ากับคีย์แพดทั้ง 7 เส้น คือสาย C0-C2 และ R0-R3 โดยที่ขาพอร์ต P2.0-P2.3 ต่อตัวต้านทานพูลอัปขนาด 10k ไว้เพื่อกำหนดสภาวะเริ่มต้นที่ไม่มีการกดคีย์

3.3 การออกแบบทางด้านซอร์ฟแวร์

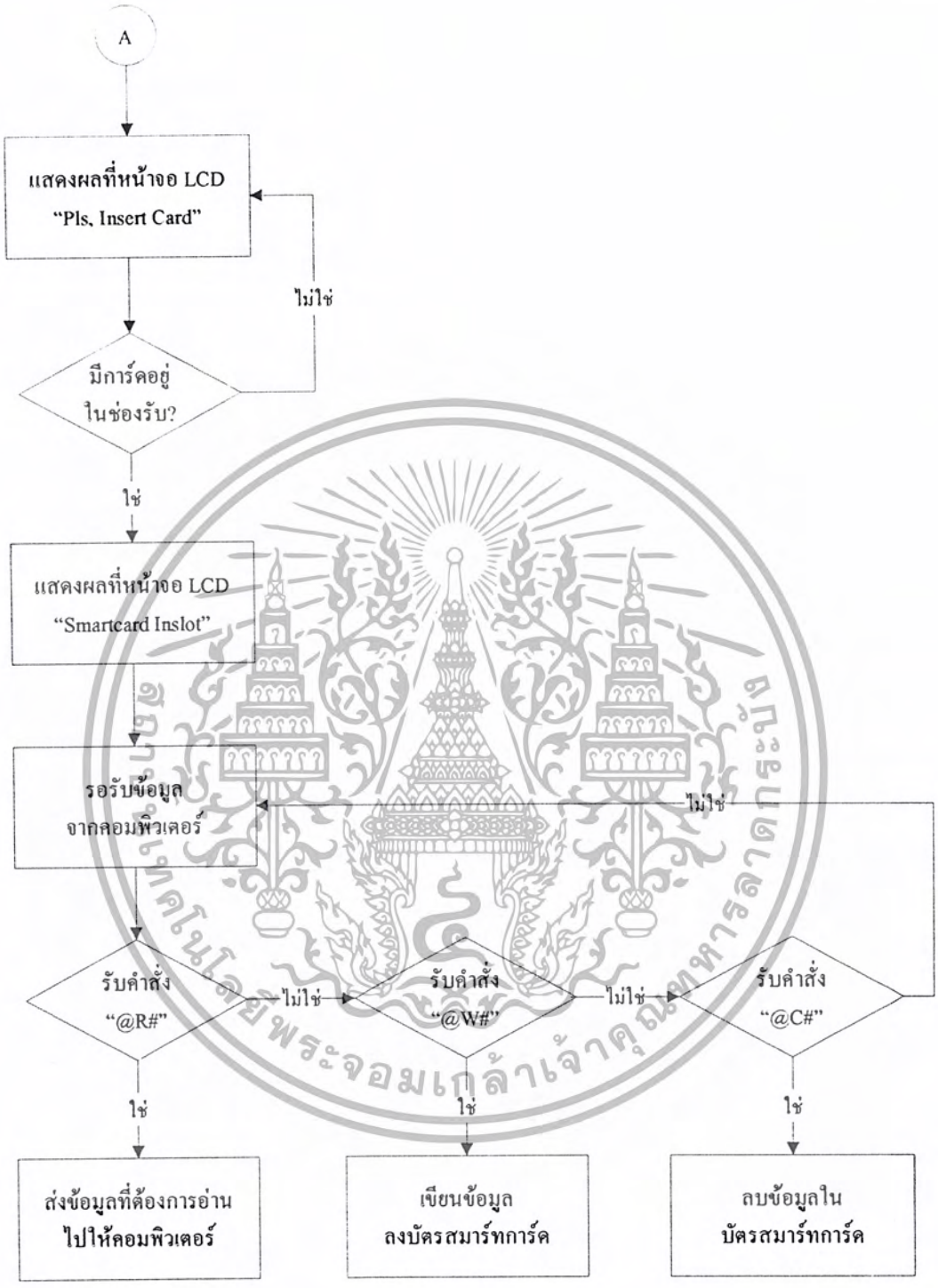
3.3.1 โปรแกรมควบคุมการทำงานของเครื่องอ่านเขียนสมาร์ตการ์ด

จากรูปที่ 3-3 และ 3-4 นั้นเป็นโฟลว์ชาร์ตของโปรแกรมควบคุมการทำงานของเครื่องอ่าน-เขียนสมาร์ตการ์ด ในขั้นตอนแรกนั้น เราต้องทำการตั้งค่าการสื่อสารของพอร์ตอนุกรม ซึ่งจะต้องตั้งค่าภายในตัวไมโครคอนโทรลเลอร์ เพื่อให้สามารถทำการติดต่อสื่อสารกับคอมพิวเตอร์ได้ และเราต้องทำการตั้งค่าจอแสดงผลแบบผลึกเหลว (LCD) ให้ทำงานตามที่เราต้องการก่อน โดยเริ่มแรกหน้าจอ LCD จะแสดงผล "PASSWD :" เพื่อให้ผู้ใช้ทำการใส่รหัสผ่านก่อนการใช้งาน ซึ่งเรียกว่ารหัส PSC หากว่าใส่รหัสผ่านไม่ถูกต้อง จะแสดงผลที่หน้าจอ LCD "PASSWORD ERROR!!" และจะมีการเช็คว่าได้ใส่รหัสผิดครบ 3 ครั้ง หรือไม่ หากครบแล้ว สมาร์ตการ์ดใบนั้นจะถูก block หมายความว่า เราไม่สามารถเขียนข้อมูลลงสมาร์ตการ์ดใบนั้นได้อีก โดยค่าที่ใช้ในการเช็คนี้ เรียกว่าค่า Error Counter ซึ่งจะมีค่าเริ่มต้นเท่ากับ 3 และจะลดลง 1 ค่าเรื่อยๆเมื่อใส่รหัสผ่านไม่ถูกต้อง ซึ่งหากลดลงจนเหลือ 0 (ป้อนรหัสผิดมาแล้ว 3 ครั้ง) แสดงว่าสมาร์ตการ์ดใบนั้นถูก block ไปเรียบร้อยแล้ว ส่วนในกรณีที่ป้อนรหัสผิดมาแล้ว 2 ครั้ง และป้อนรหัสถูกในครั้งที่ 3 ค่าของ Error Counter จะถูกรีเซตกลับไปมีค่าเท่ากับ 3 เหมือนเดิม

หากผู้ใช้ได้ป้อนรหัสที่ถูกต้องแล้ว ก็จะเข้าสู่สภาวะเตรียมพร้อม ในสภาวะเตรียมพร้อมนี้ที่จอแสดงผล LCD จะแสดงข้อความว่า "Pls, Insert Card" ซึ่งในขณะเดียวกันก็จะคอยตรวจสอบว่ามีบัตรสมาร์ตการ์ดเสียบเข้ามาที่ช่องเสียบบัตรหรือไม่ ถ้ายังไม่มีบัตรสมาร์ตการ์ดเสียบเข้ามา ที่จอแสดงผล LCD ก็จะแสดงข้อความว่า "Pls, Insert Card" ค้างไว้ แต่ถ้าหากมีบัตรสมาร์ตการ์ดเสียบเข้ามา เครื่องอ่าน-เขียนสมาร์ตการ์ดก็จะแสดงผลที่หน้าจอแสดงผลแบบผลึกเหลวว่า "Smartcard Inslot" จากนั้นจะรอรับข้อมูลจากคอมพิวเตอร์ ถ้าหากข้อมูลที่ได้รับเป็นคำสั่งในการอ่านข้อมูล ก็จะปรากฏข้อมูลต่างๆในบัตรซึ่งเปิดโดยโปรแกรมที่ติดต่อกับผู้ใช้ หากได้รับคำสั่งในการเขียนข้อมูลก็จะทำการเขียนข้อมูลลงไปในบัตรสมาร์การ์ดและถ้าได้รับคำสั่งเคลียร์ ข้อมูลทั้งหมดในบัตรจะถูกลบออกไป



รูปที่ 3-3 แสดงโฟลว์ชาร์ทของโปรแกรมควบคุมการทำงานของเครื่องอ่านเขียนสมาร์ตการ์ด



รูปที่ 3-4 แสดงโฟลว์ชาร์ทของโปรแกรมควบคุมการทำงานของเครื่องอ่านเขียนสมาร์ตการ์ด (ต่อ)

3.3.2 โปรแกรมการอ่านข้อมูลในบัตรสมาร์ตการ์ด

จากรูปที่ 3-5 เป็นโฟลวชาร์ตของโปรแกรมการอ่านข้อมูลจากบัตรสมาร์ตการ์ดและแสดงผลทางจอแสดงผลแบบผลึกเหลว ในสภาวะแรกต้องทำการส่งข้อมูลซึ่งเป็นรหัสคำสั่ง คือ สัญญาณ "@R#" เพื่อบอกให้สมาร์ตการ์ดทราบว่าต้องการอ่านข้อมูลจากหน่วยความจำหลัก จากนั้นจะต้องส่งค่าแอดเดรสที่เราต้องการอ่านข้อมูลไปด้วย ซึ่งการอ่านข้อมูลนั้นจะสามารถอ่านได้มากน้อยเพียงไหนขึ้นอยู่กับจำนวนสัญญาณนาฬิกา โดยที่ 1 แอดเดรสจะใช้สัญญาณนาฬิกา 8 ลูก จากนั้นจะทำการอ่านข้อมูลจากบัตร พร้อมแสดงผลที่หน้าจกแสดงผล LCD ว่า "Read Data..." แล้วนำมาเก็บไว้ใน Buffer ที่กำหนดไว้ และส่งข้อมูลที่อ่านได้นี้ไปแสดงผลบนคอมพิวเตอร์ต่อไป

รูปที่ 3-5 แสดงโฟลวชาร์ทของโปรแกรมการอ่านข้อมูลและแลดงผลทางจอแสดงผลแบบแอลซีดี

3.3.3 โปรแกรมการเขียนข้อมูลลงในบัตรสมาร์ตการ์ด

จากรูปที่ 3-6 แสดงโฟลวชาร์ตการทำงานของโปรแกรมการเขียนข้อมูลลงบัตรสมาร์ตการ์ด ไมโครคอนโทรลเลอร์จะเป็นตัวส่งคำสั่งการเขียนข้อมูลไปยังบัตรสมาร์ตการ์ด ซึ่งเป็นสัญญาณ "@W#" จากนั้นจะส่งค่าแอดเดรสที่ต้องการเขียนข้อมูลไปยังบัตร และทำการส่งข้อมูลที่ต้องการเขียนลงในบัตร จนกระทั่งมีขนาดครบตามที่กำหนดไว้ จึงได้รับสัญญาณตอบรับ และทำการเขียนข้อมูลลงในบัตร พร้อมกับหน้าจอแสดงผล LCD จะแสดงผล "Write Data..." โดยสัญญาณตอบรับดังกล่าวกำหนดเป็นสัญญาณ "#" ซึ่งมีความหมายคล้ายๆกับการส่งข้อมูลได้เสร็จสิ้นไปแล้ว 1 ชุด จากนั้นหากต้องการส่งข้อมูลชุดต่อไป ก็จะส่งต่อไปจนได้รับสัญญาณตอบรับอีกครั้งหนึ่ง ทำอย่างนี้ไปเรื่อยๆจนกระทั่งส่งข้อมูลได้ครบตามที่ต้องการ

รูปที่ 3-6 แสดงโฟลวชาร์ทการทำงานของโปรแกรมการเขียนข้อมูลลงในบัตรสมาร์ตการ์ด

3.3.4 โปรแกรมการลบข้อมูลในบัตรสมาร์ตการ์ด

จากรูปที่ 3-7 เป็นโฟลวชาร์ตของโปรแกรมการลบข้อมูลในบัตรสมาร์ตการ์ด ในสภาวะแรกต้องทำการส่งข้อมูลซึ่งเป็นรหัสคำสั่ง คือสัญญาณ "@C#" เพื่อบอกให้สมาร์ตการ์ดทราบว่าต้องการลบข้อมูลทั้งหมดจากหน่วยความจำหลัก หลังจากที่ได้ส่งสัญญาณไปแล้ว หน้าจอแสดงผล LCD จะแสดงผล "Clear Data..." พร้อมกับลบข้อมูลทั้งหมดในบัตรสมาร์ตการ์ด

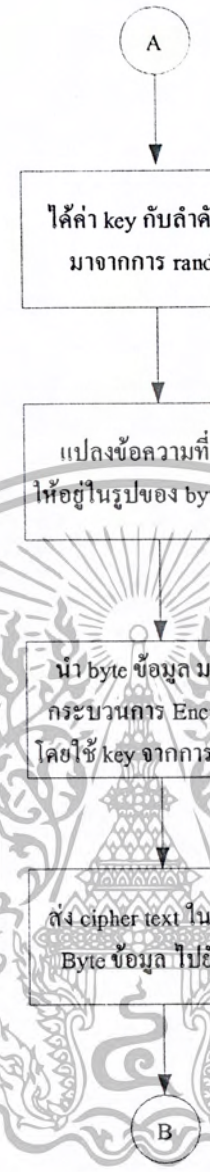
รูปที่ 3-7 แสดงโฟลวชาร์ทการทำงานของโปรแกรมการลบข้อมูลในบัตรสมาร์ตการ์ด

3.3.5 โปรแกรมการส่งข้อมูลเข้ารหัสลับและถอดรหัสลับ

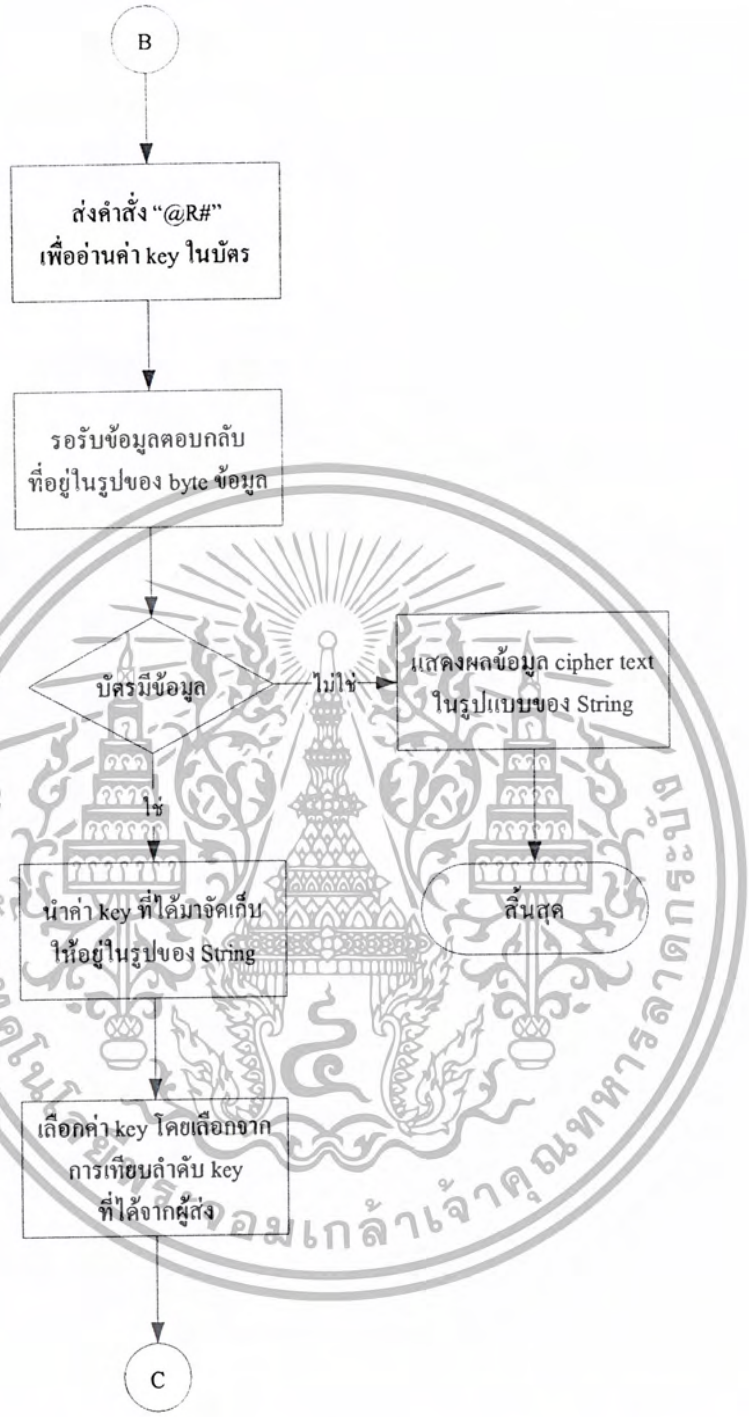
จากรูป 3-8 ถึง 3-11 นั้น แสดงถึงโฟลวชาร์ตของโปรแกรมการส่งข้อมูลที่ถูกเข้ารหัสลับ (Cipher text) และข้อมูลดังกล่าวจะถูกถอดรหัสลับเมื่อถึงผู้รับ โดยในรูปที่ 3-8 และ 3-9 นั้น แสดงกระบวนการทำงานของทางด้านฝั่งผู้ส่ง และในรูป 3-10 และ 3-11 นั้น แสดงกระบวนการทำงานของทางด้านฝั่งผู้รับ โดยขั้นตอนแรกนั้น เมื่อเรากดปุ่มส่งข้อมูลแบบ Cipher text แล้ว โปรแกรมจะส่งสัญญาณในการอ่านข้อมูลในบัตรสมาร์ตการ์ด คือสัญญาณ "@R#" แล้วโปรแกรมจะรอข้อมูลตอบรับจากเครื่องอ่านสมาร์ตการ์ด โดยจะเช็คว่ามีข้อมูล หรือค่าของกุญแจรหัส (Secret key) อยู่ในบัตร หรือมีบัตรเสียบอยู่ที่เครื่องอ่านสมาร์ตการ์ดหรือไม่ หากมีจะส่งค่ากุญแจรหัสกลับมายังโปรแกรม ซึ่งจะอยู่ในรูปของไบต์ข้อมูล แล้วโปรแกรมจะจัดเก็บค่ากุญแจรหัสดังกล่าว โดยจัดเก็บให้อยู่ในตัวแปรในรูปของสตริง (String) แล้วจึงสามารถเข้าสู่อัลกอริทึมในการสุ่มเลือกค่ากุญแจรหัสจากชุดของสตริงที่ได้จากขั้นตอนดังกล่าว เมื่อเลือกค่ากุญแจรหัสพร้อมกับลำดับของกุญแจรหัสเสร็จแล้ว จะนำค่ากุญแจรหัสที่ได้มาแปลงเป็นไบต์ พร้อมกับนำข้อความที่ผู้ส่งต้องการจะเข้ารหัสลับมาแปลงค่าให้เป็นไบต์ข้อมูล ก่อนจะนำไปเข้าสู่กระบวนการ encryption โดยใช้กุญแจรหัสที่ได้สุ่มเลือกขึ้นมา แล้วส่งข้อมูล cipher text ที่อยู่ในรูปของไบต์ข้อมูล พร้อมกับลำดับของกุญแจรหัสไปยังปลายทางต่อไป

ทางด้านผู้รับ เมื่อได้รับข้อมูล Cipher text พร้อมกับลำดับของกุญแจรหัสดังกล่าวแล้ว โปรแกรมจะส่งสัญญาณ "@R#" เพื่อทำการอ่านค่าของกุญแจรหัสที่เก็บอยู่ในบัตรสมาร์ตการ์ด ในขั้นตอนนี้จะเช็คว่าในบัตรของผู้รับมีข้อมูลหรือไม่ หากไม่มี หรือไม่ได้เสียบบัตร ก็จะไม่เข้าสู่กระบวนการ Decryption และทำการแสดงผลข้อมูล cipher text ที่อ่านไม่รู้เรื่องให้กับผู้รับ ซึ่งเป็นการแจ้งเตือนผู้รับอีกทางหนึ่ง ว่าบัตรสมาร์ตการ์ดของตนเองไม่สมบูรณ์ หรือลืมเสียบบัตรเข้ากับเครื่องอ่านสมาร์ตการ์ด แต่หากมีข้อมูลในบัตร จึงจะทำการอ่านค่ามาในรูปของไบต์ข้อมูลมาเก็บไว้ในตัวแปรในรูปของสตริง แล้วจึงจะนำลำดับของกุญแจรหัส มาเทียบกับชุดของกุญแจรหัส ที่อยู่ในรูปของสตริงแล้วจึงได้ค่ากุญแจรหัสที่ถูกต้อง และนำค่ากุญแจรหัสมาแปลงให้อยู่ในรูปของไบต์ข้อมูล พร้อมกับนำไบต์ข้อมูล cipher text ที่ได้รับ มาเข้าสู่กระบวนการ decryption โดยใช้ค่ากุญแจรหัสที่ได้ แล้วจึงได้ข้อมูลประเภท clear text โดยมีค่าของข้อมูลเหมือนกับข้อมูลของผู้ส่งที่ก่อนจะนำมาเข้ารหัสลับทุกประการ จากนั้นจึงแสดงผลข้อมูลที่ถูกถอดรหัสลับแล้วทางด้านฝั่งผู้รับต่อไป

รูปที่ 3-8 แสดงโฟลวชาร์ทการทำงานของโปรแกรมการส่งข้อมูลเข้ารหัสลับและถอดรหัสลับ



รูปที่ 3-9 แสดงโฟลวชาร์ทการทำงานของโปรแกรมการส่งข้อมูลเข้ารหัสลับและถอดรหัสลับ (ต่อ)



รูปที่ 3-10 แสดงโฟลวชาร์ทการทำงานของโปรแกรมการส่งข้อมูลเข้ารหัสลับและถอดรหัสลับ (ต่อ)

รูปที่ 3-11 แสดงโฟลวชาร์ทการทำงานของโปรแกรมการส่งข้อมูลเข้ารหัสลับและถอดรหัสลับ (ต่อ)

บทที่ 4

ผลการทดลอง

4.1 ผลการทดลองในส่วนของโปรแกรมการบันทึกและอ่านข้อมูลจากบัตรสมาร์ตการ์ด

4.1.1 เริ่มต้นการทำงาน

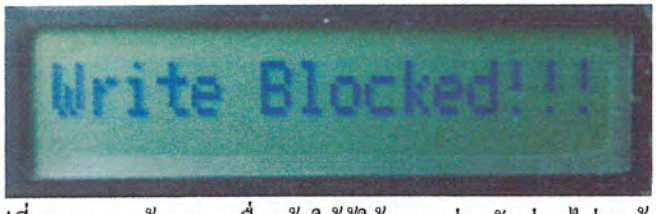
- เครื่องอ่าน-เขียนสมาร์ตการ์ดแสดงข้อความ “PASSWD :” เพื่อให้ผู้ใช้ทำการใส่รหัสผ่านสำหรับการใช้งานเครื่องอ่าน-เขียนสมาร์ตการ์ดเครื่องนี้

รูปที่ 4-1 แสดงข้อความเพื่อให้ผู้ใช้ทำการใส่รหัสผ่าน

- ถ้าผู้ใช้ใส่รหัสผ่านไม่ถูกต้อง จอแอลซีดีจะแสดงข้อความ “PASSWORD ERROR!!” ให้ผู้ใช้ทำการใส่รหัสผ่านใหม่

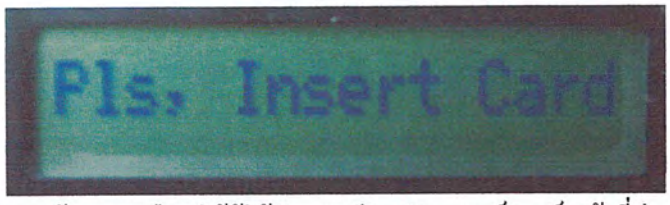
รูปที่ 4-2 แสดงข้อความเพื่อแจ้งให้ผู้ใช้ทราบว่ารหัสผ่านไม่ถูกต้อง

- ในกรณีที่ใส่รหัสผ่านไม่ถูกต้องติดกันถึงสามครั้ง จอแอลซีดีจะแสดงข้อความ “Write Blocked!!!” เพื่อแสดงให้ผู้ใช้ทราบว่าสมาร์ตการ์ดใบนี้จะไม่สามารถทำการบันทึกข้อมูลได้



รูปที่ 4-3 แสดงข้อความเพื่อแจ้งให้ผู้ใช้ทราบว่ารหัสผ่านไม่ถูกต้อง

- เครื่องอ่าน-เขียนสมาร์ตการ์ดแสดงข้อความ “Pls, Insert Card” แจ้งเตือนให้ผู้ใช้ทำการเสียบบัตรสมาร์ตการ์ดไปที่ช่องสล็อตสมาร์ตการ์ด

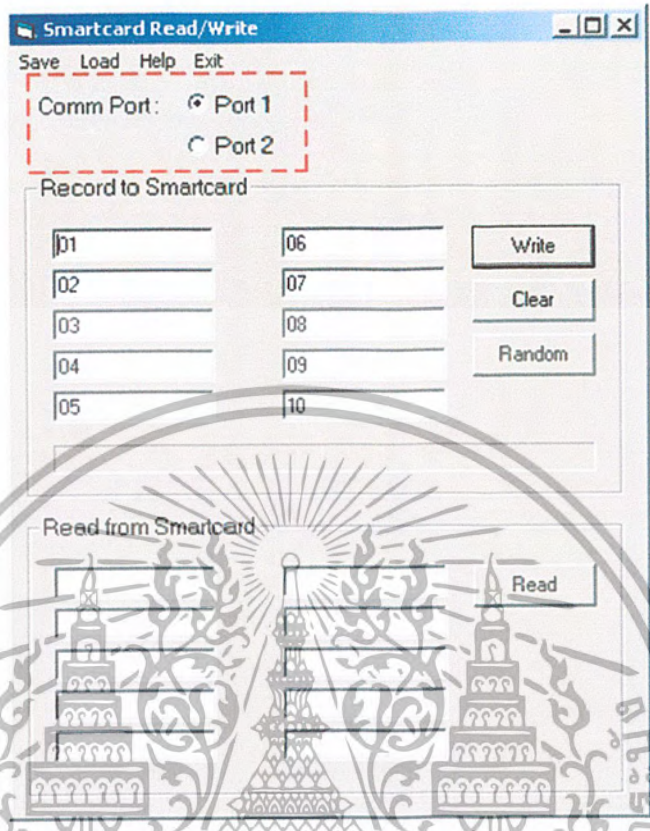


รูปที่ 4-4 แสดงข้อความเตือนให้ผู้ใช้ทำการเสียบบัตรสมาร์ตการ์ดเข้าที่ช่องเสียบบัตร

- เมื่อบัตรสมาร์ตการ์ดถูกเสียบเข้าไปที่ช่องเสียบบัตรแล้ว จอแอลซีดีจะแสดงข้อความ “Smartcard Inslot” เป็นการเตรียมพร้อมสู่ขั้นตอนการอ่านหรือเขียนข้อมูล

รูปที่ 4-5 แสดงข้อความเตรียมพร้อมสู่ขั้นตอนการบันทึกหรืออ่านข้อมูล

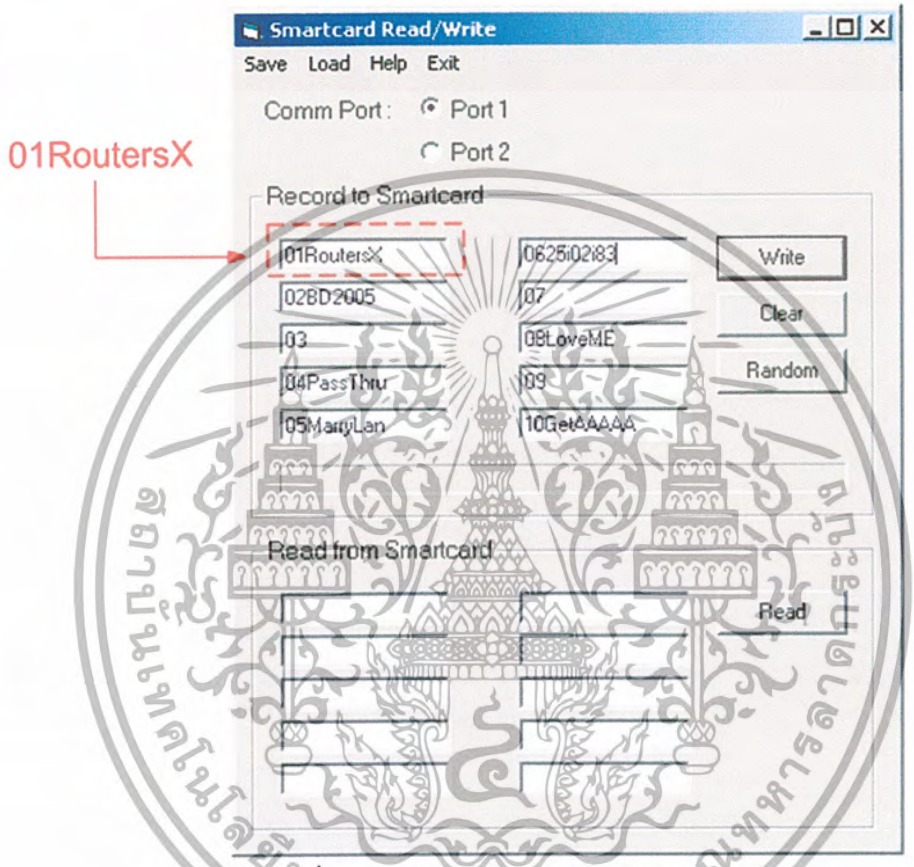
- เปิดโปรแกรม Smartcard Read/Write ขึ้นมา จากนั้นให้ทำการเลือกพอร์ตที่ใช้ในการเชื่อมต่อกับเครื่องอ่าน-เขียนสมาร์ตการ์ดเป็นอันดับแรก ซึ่งในการทดลองนี้จะเลือกพอร์ตที่ 1



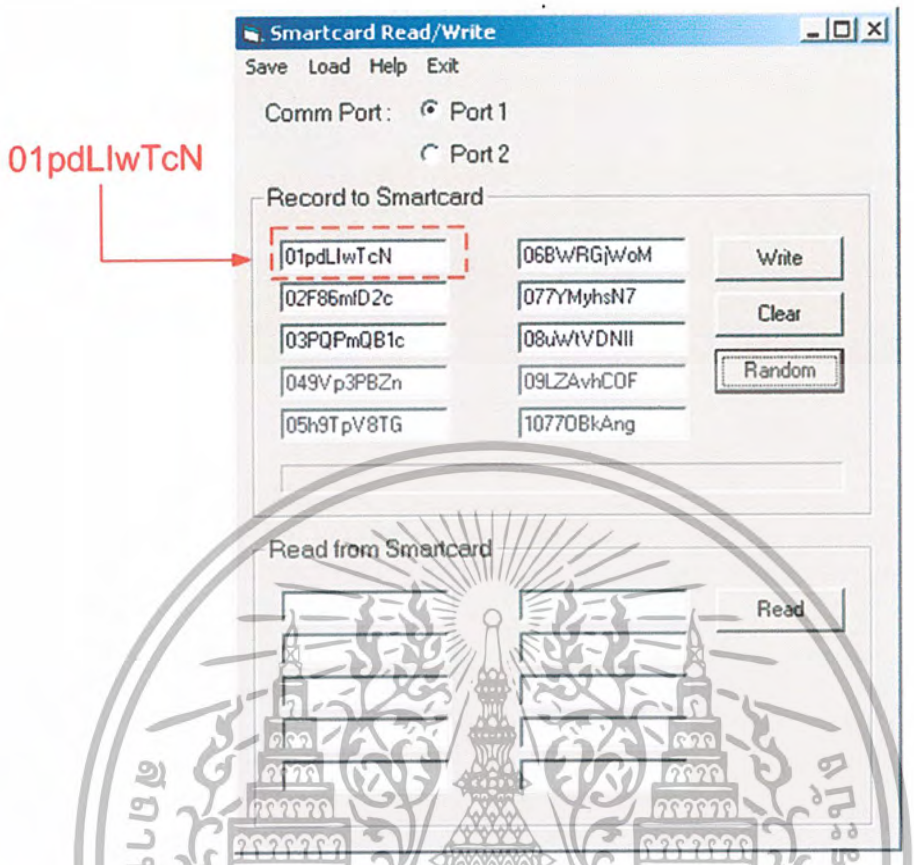
รูปที่ 4-6 ทำการเลือกพอร์ตที่ใช้ในการเชื่อมต่อกับเครื่องอ่าน-เขียนสมาร์ตการ์ด

4.1.2 การเขียนข้อมูล

- ข้อมูลที่บันทึกลงไปในสมาร์ตการ์ด คือ กุญแจรหัส ซึ่งในการทดลองนี้ จะกำหนดให้มีทั้งหมด 10 ค่า หรือ 10 ชุด ซึ่งสามารถทำการเขียนได้สองวิธี คือ การใส่ค่าลงไปในช่องว่างด้วยตนเอง หรืออีกวิธีหนึ่งโดยการกดปุ่ม Random เพื่อทำการสุ่มค่า



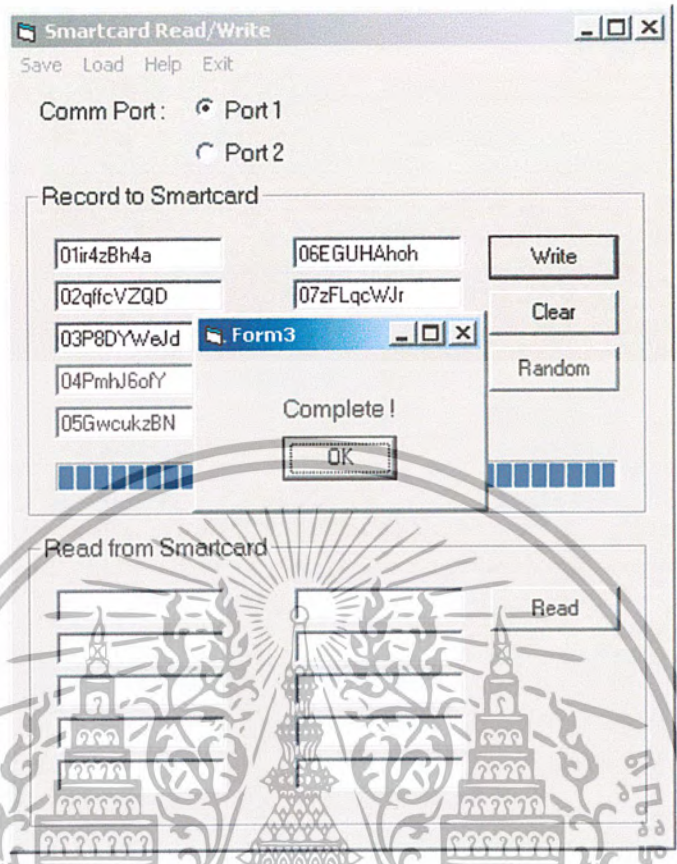
รูปที่ 4-7 การกำหนดค่าของกุญแจรหัสด้วยตนเอง



รูปที่ 4-8 การกำหนดค่าของกุญแจรหัสด้วยการสุ่ม

- เมื่อกำหนดค่าของกุญแจรหัสแล้ว กดปุ่ม Write เพื่อบันทึกข้อมูลลงในบัตร ในขณะที่กำลังบันทึกข้อมูล หน้าจอแอลซีดีจะแสดงข้อความ "Write Data. . ." ส่วนหน้าต่างของโปรแกรมจะแสดงแถบความก้าวหน้า (Progress Bar) และหน้าต่าง complete เพื่อแจ้งให้ผู้ใช้ทราบเมื่อการบันทึกข้อมูลเสร็จสิ้นแล้ว

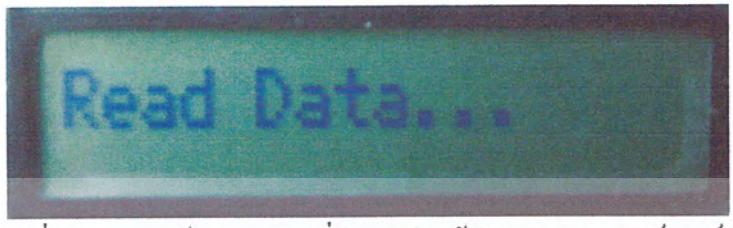
รูปที่ 4-9 แสดงข้อความขณะที่ทำการบันทึกข้อมูลลงในสมาร์ตการ์ด



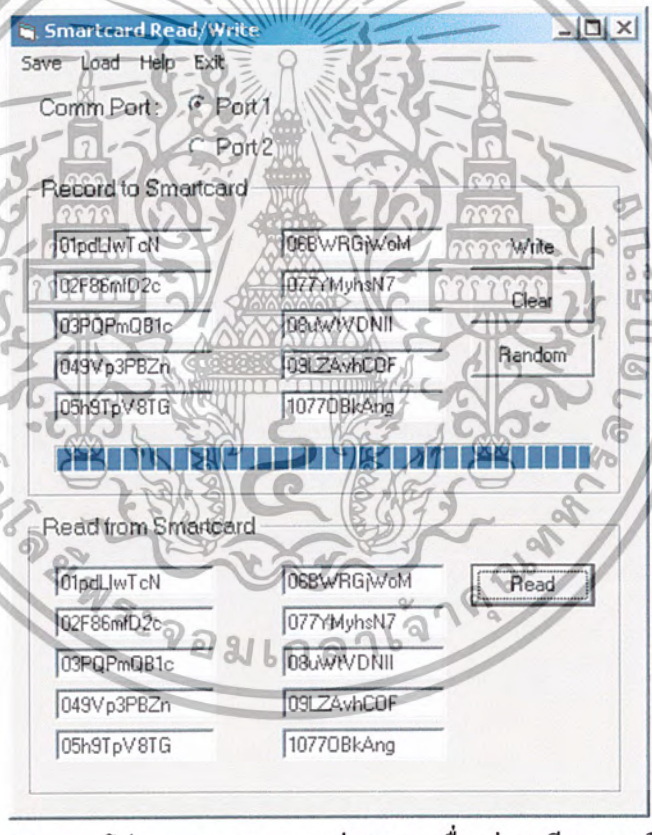
รูปที่ 4-10 แสดงแถบความก้าวหน้าและหน้าต่าง Complete

4.1.3 การอ่านข้อมูล

- เมื่อกดปุ่ม Read จะเป็นการอ่านค่ากุญแจรหัสจากบัตรสมาร์ตการ์ดให้มาแสดงที่หน้าต่างของโปรแกรม และในขณะที่ทำการอ่านข้อมูล หน้าจอแอลซีดีจะแสดงข้อความ “Read Data. . . ”



รูปที่ 4-11 แสดงข้อความขณะที่ทำการอ่านข้อมูลจากบัตรสมาร์ตการ์ด



รูปที่ 4-12 แสดงโปรแกรมควบคุมการอ่านของเครื่องอ่าน-เขียนสมาร์ตการ์ด

4.1.4 การลบข้อมูล

- เมื่อกดปุ่ม Clear จะเป็นการลบค่ากุญแจรหัสทั้งหมดที่อยู่ในบัตร ที่หน้าจอแอลซีดีจะแสดงข้อความ “Clear Data. . .”

รูปที่ 4-13 แสดงข้อความขณะที่ทำการอ่านข้อมูลจากบัตรสมาร์ตการ์ด

4.1.5 การจบการทำงาน

- เมื่อต้องการจบการทำงานของโปรแกรม ให้กดที่ปุ่ม Exit

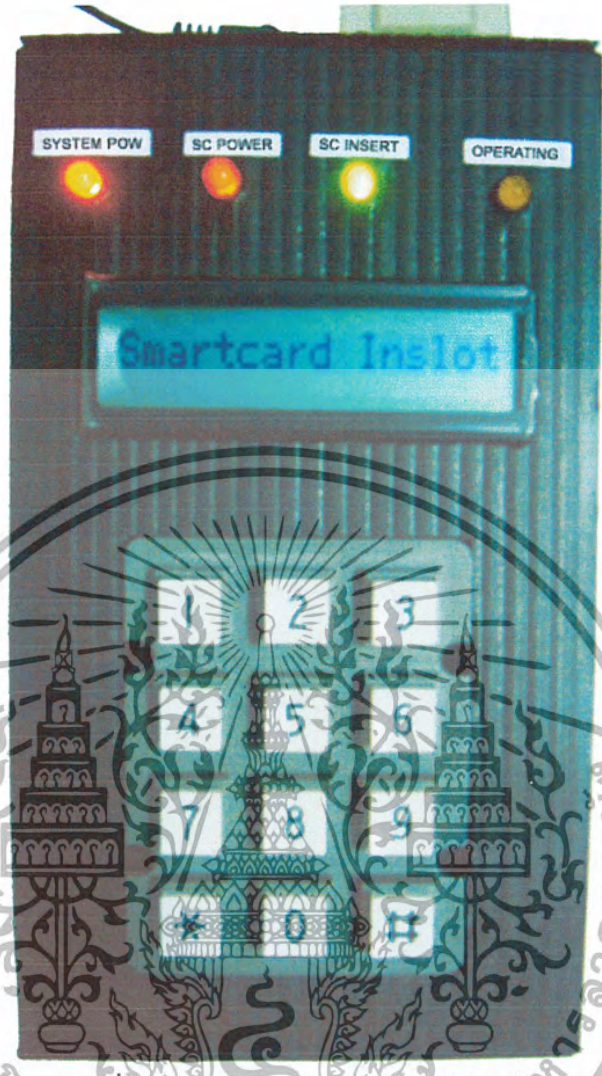
4.1.6 สถานะของหลอดไฟแอลอีดี (LED)

- LED1: System Power แสดงถึงสถานะไฟเลี้ยงของระบบ หลอดไฟสีแดงจะติดเมื่อต่อเครื่องอ่าน-เขียนสมาร์ตการ์ดเข้ากับอแดปเตอร์ที่ต่อกับปลั๊กไฟ

- LED2: Smartcard Power แสดงถึงสถานะไฟเลี้ยงของสมาร์ตการ์ด หลอดไฟสีแดงจะติดเมื่อสมารทการ์ดถูกเสียบเข้ามาในช่องเสียบ

- LED3: Smartcard Insert แสดงถึงการเชื่อมต่อระหว่างการ์ดกับช่องเสียบ หลอดไฟสีเขียวจะติดเมื่อการ์ดถูกเสียบเข้าไปในช่องเสียบ

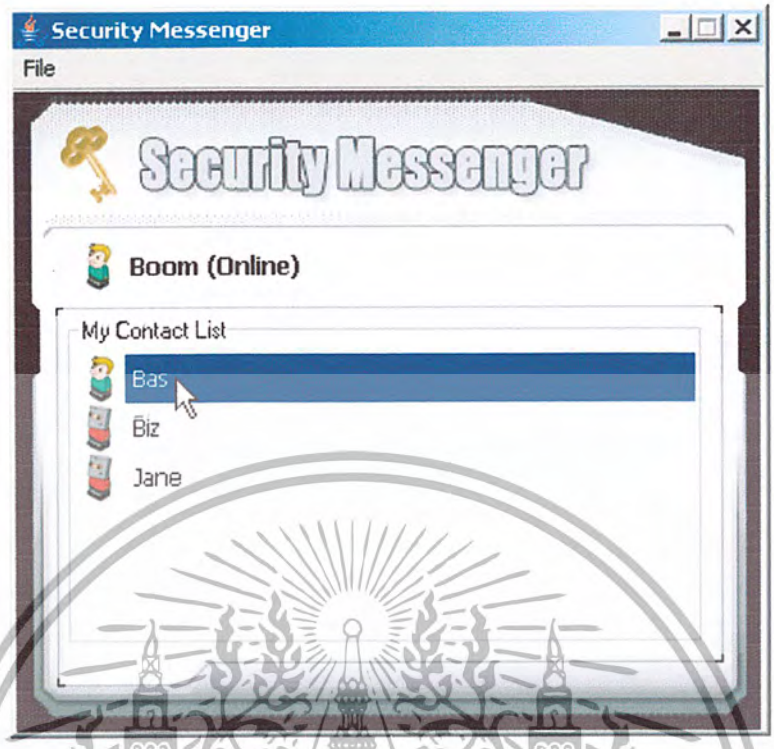
- LED4: Operating แสดงถึงการทำงานของสมาร์ตการ์ด หลอดไฟสีเหลืองจะติด เมื่อมีการบันทึกหรืออ่านข้อมูล



รูปที่ 4.14 แสดงสถานะของหลอดไฟแอลอีดี

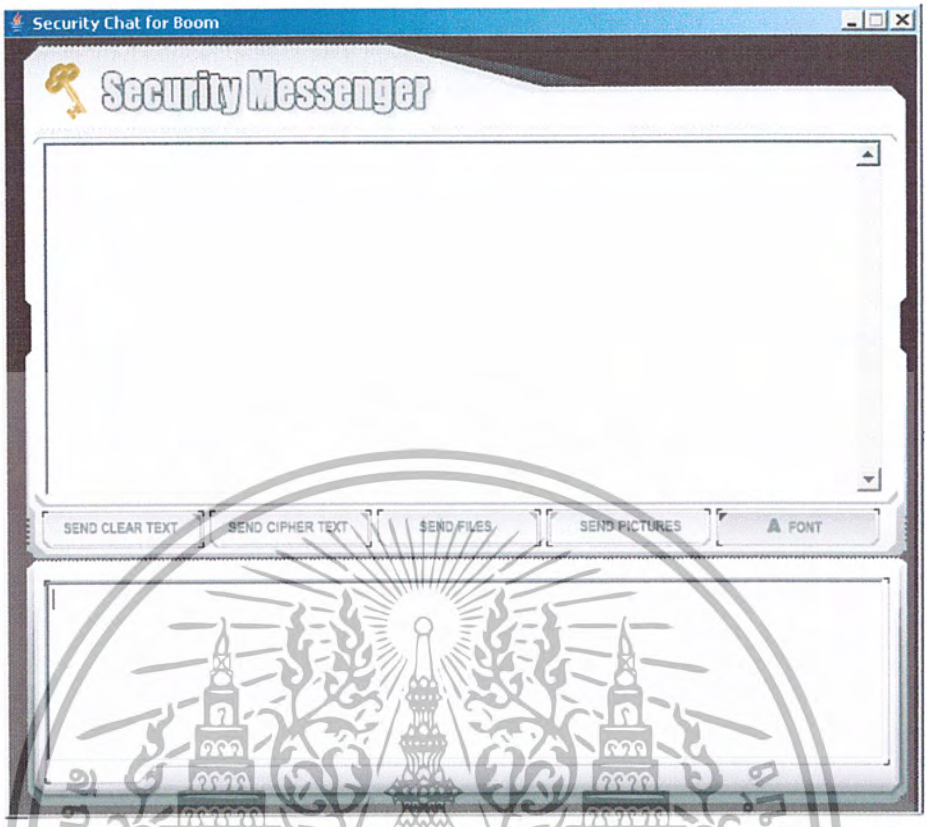
4.2 ผลการทดลองในส่วนของโปรแกรมการส่งข้อมูลแบบเข้ารหัสลับและถอดรหัสลับ

ในขั้นต้น ผู้ใช้จะต้องทำการ Sign in เข้ามาใช้งานโปรแกรมสนทนา โดยจะมีช่องให้ใส่ username และ password ซึ่งเมื่อผู้ใช้ได้ sign in เข้ามาแล้ว จะมีลักษณะของโปรแกรม ดังรูปที่ 4-15



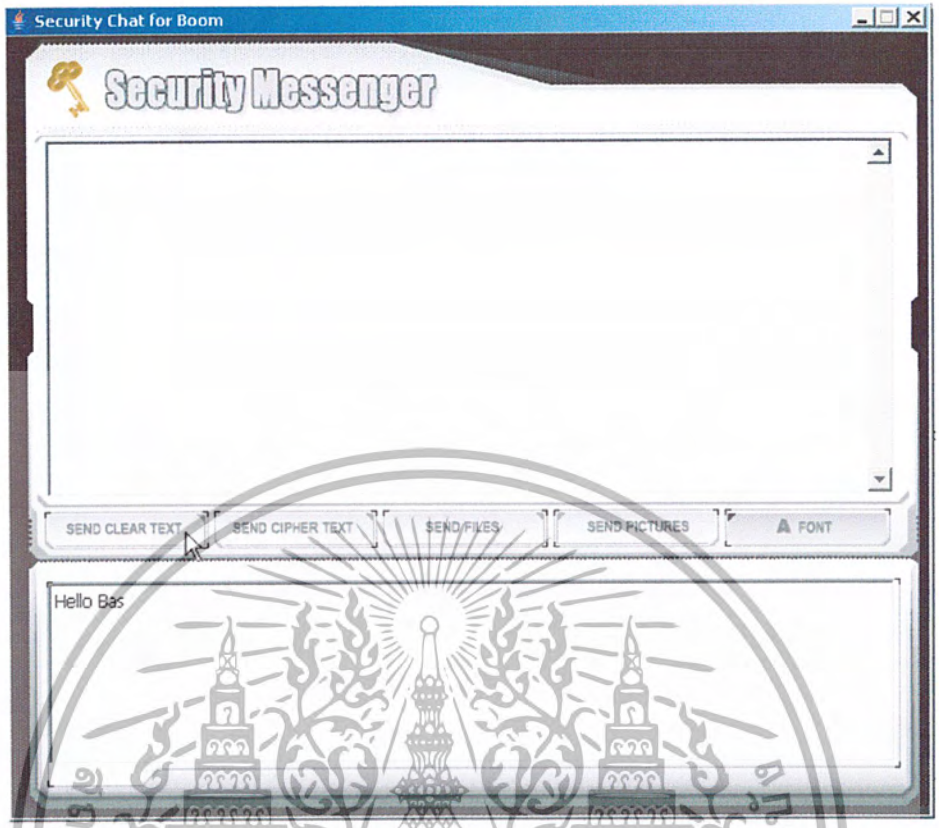
รูปที่ 4-15 แสดงถึง Contact list ซึ่งมีผู้ online อยู่

จากนั้น หากเราต้องการสนทนากับคนไหน ก็เพียงดับเบิ้ลคลิกที่ชื่อของคนนั้น จะปรากฏหน้าต่างของการสนทนาขึ้นมาให้เราใช้งาน โดยลักษณะการทำงานของหน้าต่างสนทนาของโปรแกรมนี้ เราสามารถเลือกได้ว่าต้องการส่งข้อมูลที่เป็นแบบไม่เข้ารหัสลับ (Clear text) หรือส่งข้อมูลที่ถูกเข้ารหัสลับ (cipher text) ซึ่งเป็นข้อมูลที่เป็นความลับ และไม่ต้องการให้ผู้อื่นรับรู้นอกจากคู่สนทนาของเรา ซึ่งผู้ประสงค์ร้ายที่ต้องการรับรู้ถึงข้อมูลดังกล่าว โดยอาจใช้วิธีการดักจับข้อมูล จะได้แต่ข้อมูลที่ถูกเข้ารหัสลับแล้ว โดยมีลักษณะของหน้าต่างสนทนา ดังรูปที่ 4-16

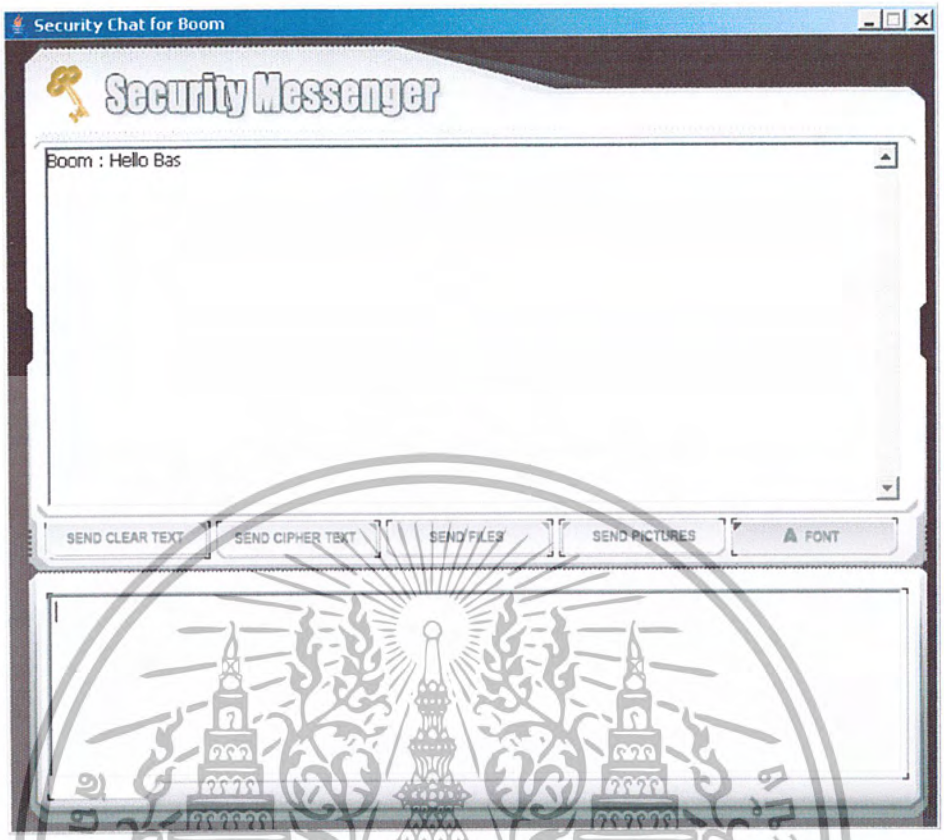


รูปที่ 4-16 แสดงถึงหน้าต่างของโปรแกรมสนทนา

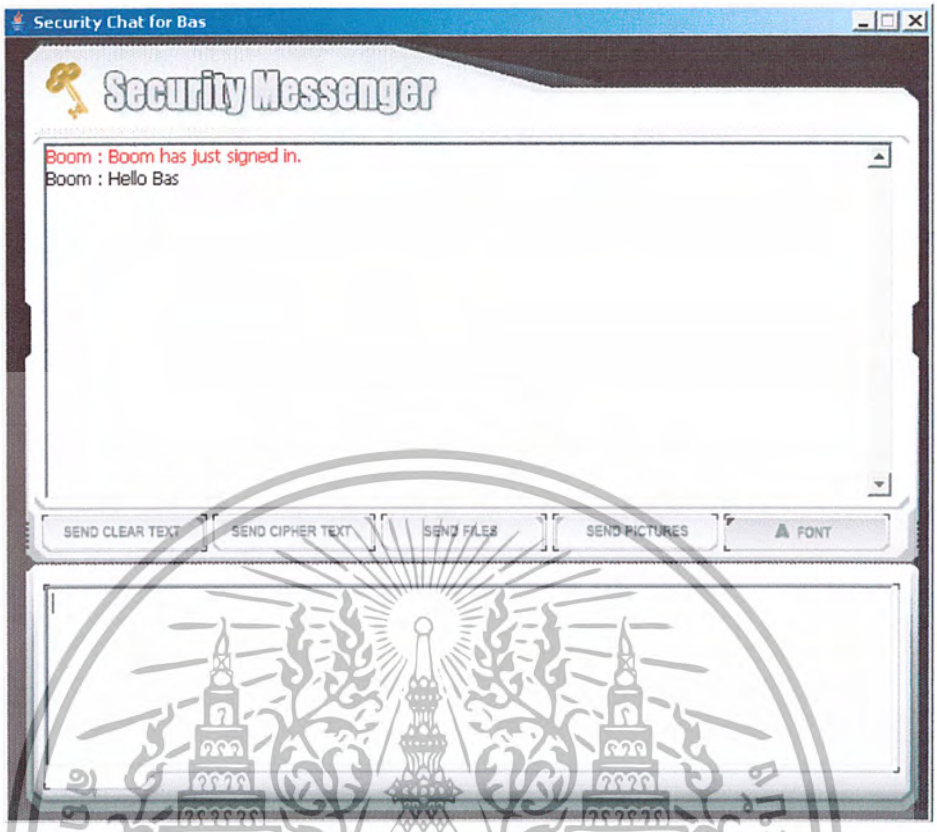
หลังจากที่พิมพ์ข้อความลงในช่องใส่ข้อความแล้ว หากข้อความดังกล่าว เป็นข้อมูลที่ไม่เป็นความลับนั้น เราจะคลิกปุ่ม "Send clear text" เพื่อส่งข้อมูลโดยไม่ต้องผ่านการเข้ารหัสลับ



รูปที่ 4-17 แสดงถึงข้อความของผู้ส่งที่ต้องการส่งแบบไม่เข้ารหัสลับ

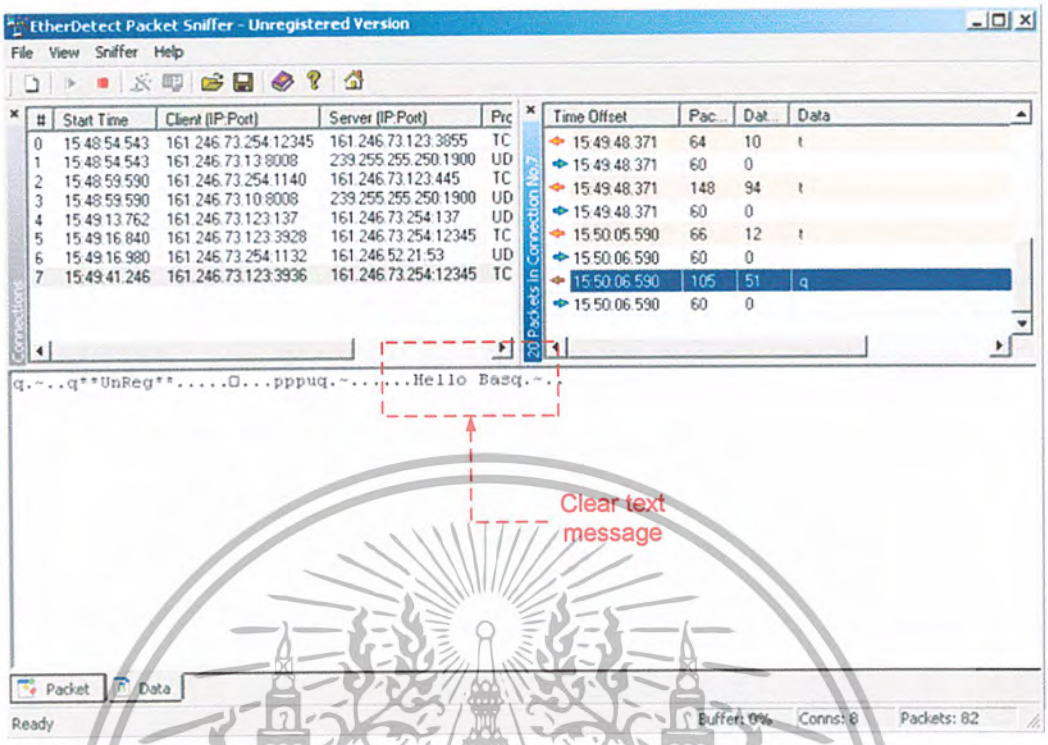


รูปที่ 4-18 แสดงถึงการแสดงผลข้อมูลประเภท clear text ทางด้านผู้ส่ง



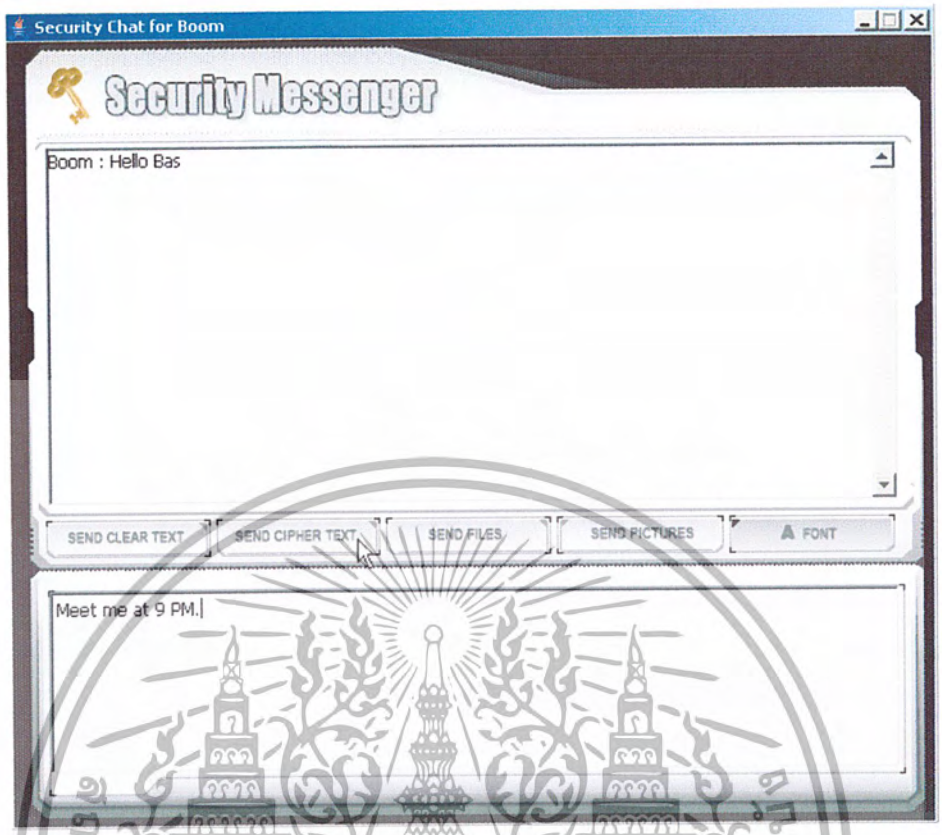
รูปที่ 4-19 แสดงถึงการแสดงผลข้อมูลประเภท clear text ทางด้านผู้รับ

จากขั้นตอนดังกล่าว เมื่อเราทำการดักจับ Packet ข้อมูลด้วยโปรแกรม EtherDetect แล้ว จะเห็นว่า ข้อมูลที่ถูกดักจับได้นั้น เราสามารถเข้าใจถึงเนื้อหาของข้อมูลได้ไม่ยาก เนื่องจากเป็นข้อมูลประเภท clear text ดังรูปที่ 4-20

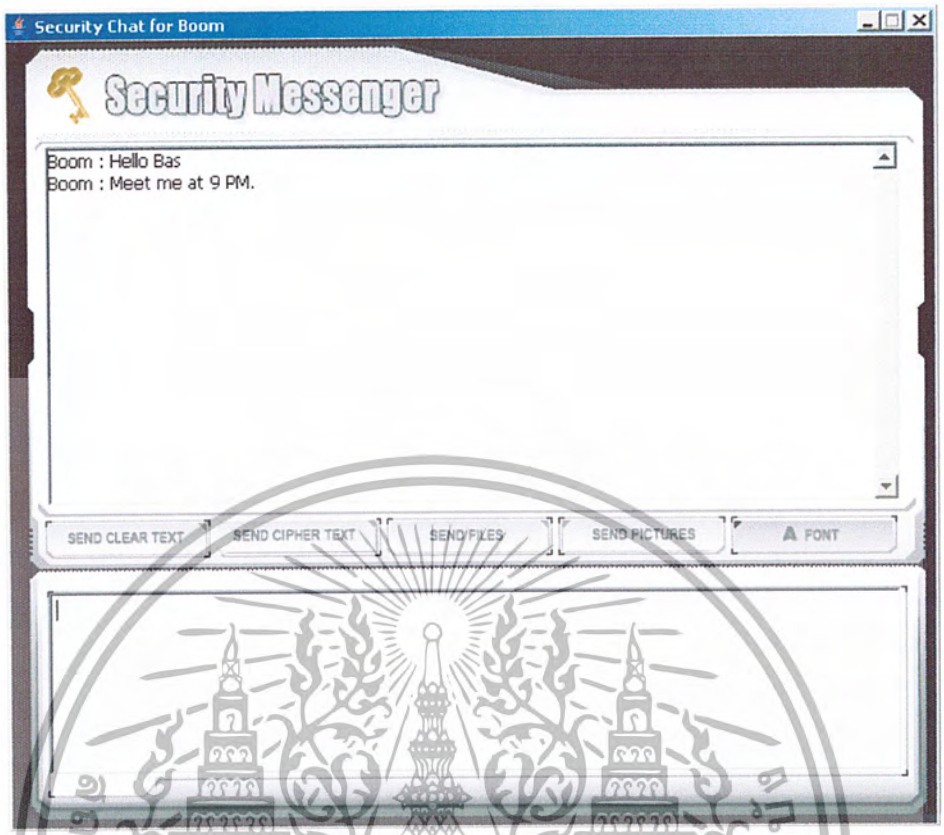


รูปที่ 4-20 แสดงถึงการใช้โปรแกรม Sniffer ในการดักจับข้อมูลประเภท clear text โดยทำการดักจับข้อมูลจากผู้ที่ส่งข้อความดังรูปที่ 4-17

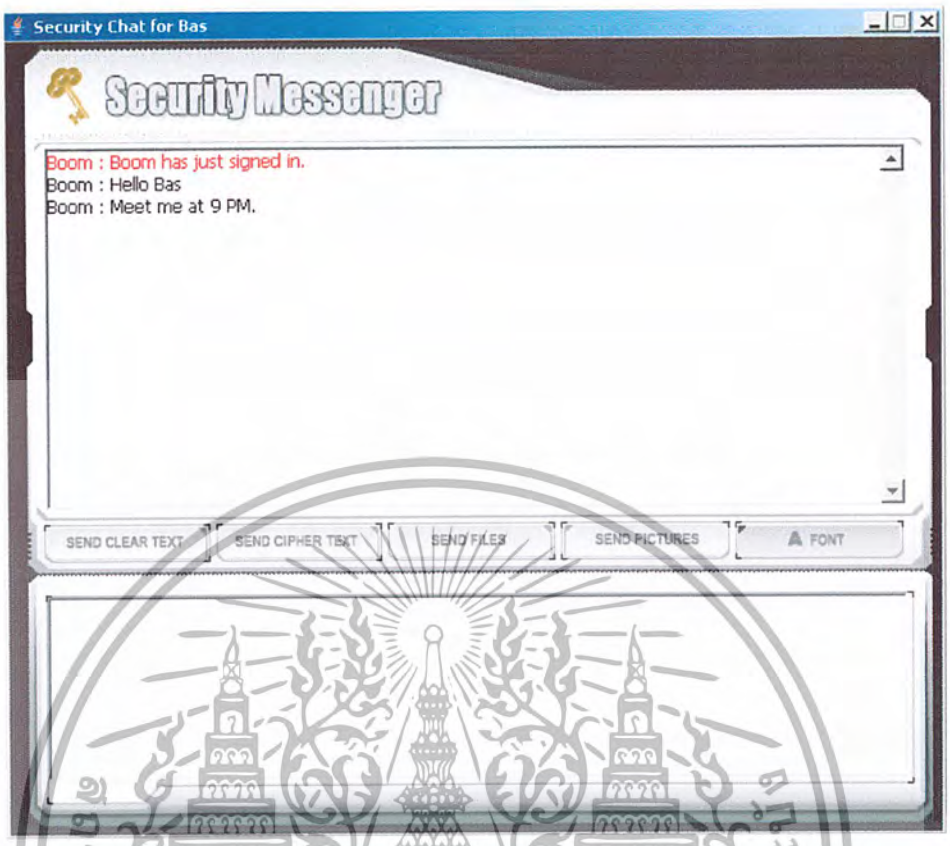
ในกรณีที่เราต้องการส่งข้อมูลที่เป็นความลับนั้น เราจะคลิกปุ่ม “Send cipher text” เพื่อทำการส่งข้อมูลที่ถูกเข้ารหัสลับ ไปยังผู้รับปลายทาง และผู้รับจะนำเอาข้อมูลที่อยู่ในรูปของ cipher text ที่ได้รับ มาเข้าสู่กระบวนการถอดรหัสลับก่อนที่จะนำไปแสดงผล เพราะฉะนั้นทางฝั่งผู้รับ จึงสามารถแสดงผลข้อมูลที่สามารถเข้าใจได้ โดยอาจไม่จำเป็นต้องรับรู้เลยว่าข้อมูลที่ถูกส่งมานั้น มีความปลอดภัยสูง เพราะได้ถูกเข้ารหัสลับเรียบร้อยแล้ว ดังรูปที่ 4-21 ถึง รูปที่ 4-23



รูปที่ 4-21 แสดงถึงข้อความของผู้ส่งที่ต้องการส่งแบบเข้ารหัสลับ

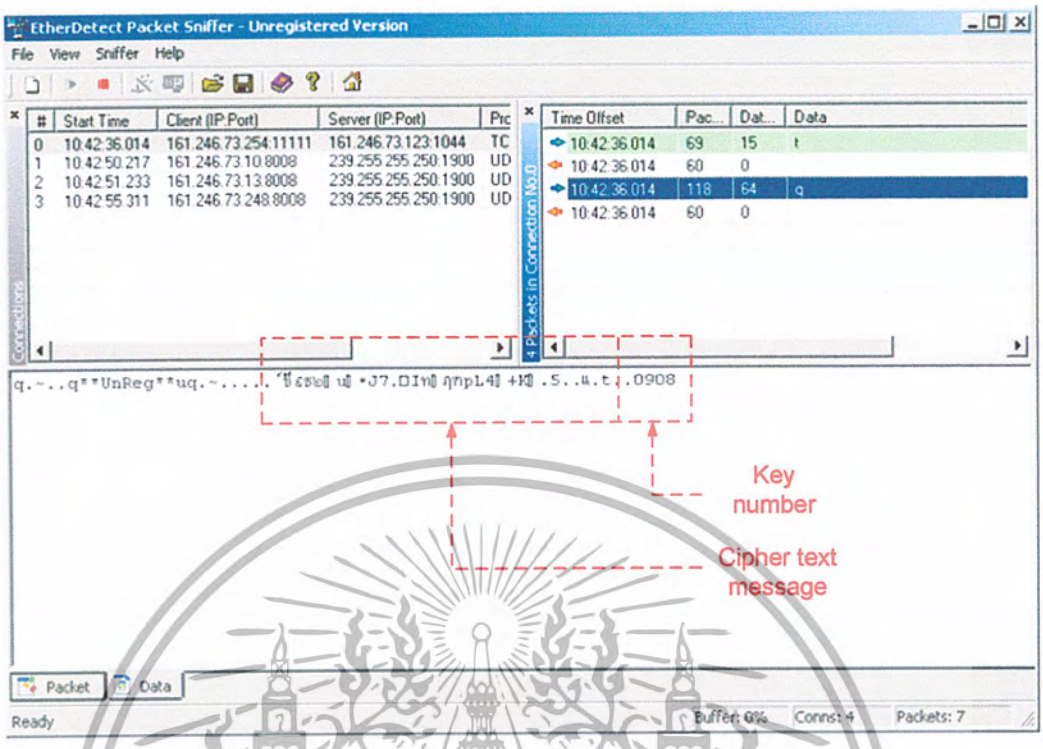


รูปที่ 4-22 แสดงถึงการแสดงผลข้อมูลประเภท cipher text ทางด้านผู้ส่ง



รูปที่ 4-23 แสดงถึงการแสดงผลข้อมูลประเภท cipher text ทางด้านผู้รับ

จากขั้นตอนดังกล่าว เมื่อเราทำการดักจับข้อมูลแล้ว จะได้ข้อมูลที่เราอ่านไม่เข้าใจ ซึ่งผู้ดักจับ หากไม่ทราบถึงอัลกอริทึม และกุญแจรหัสที่ถูกต้องแล้ว จะไม่สามารถรับรู้ถึงข้อมูลได้เลย ซึ่งจากรูปที่ 4-24 จะเห็นว่าข้อมูลที่ถูกดักจับ จะได้ข้อมูลประเภท Cipher text พร้อมกับหมายเลขลำดับของกุญแจรหัสทั้ง 2 ลำดับ เนื่องจากเป็นการเข้ารหัสลับแบบ 2 ชั้น เพื่อเพิ่มความปลอดภัย โดยใช้การเข้ารหัสลับด้วยอัลกอริทึม DES ก่อน และตามด้วยอัลกอริทึม Blowfish และยังใช้กุญแจรหัสคนละตัวอีกด้วย จึงต้องส่งลำดับของกุญแจรหัสไปยังผู้รับทั้ง 2 ลำดับ เพื่อให้ผู้รับนำลำดับดังกล่าวไปเทียบกับกุญแจรหัสทั้ง 2 ค่าของตนเอง และสามารถเข้าสู่กระบวนการถอดรหัสลับแบบ 2 ชั้นได้ (จากรูป Key number = 0908 หมายถึง กุญแจรหัสลำดับที่ 09 เป็นกุญแจรหัสของ DES และกุญแจรหัสลำดับที่ 08 เป็นกุญแจรหัสของ Blowfish) โดยข้อมูล Cipher text ที่ได้รับมา จะถูกนำมาเข้าสู่กระบวนการถอดรหัสลับด้วยอัลกอริทึม Blowfish เสียก่อน แล้วจึงถูกถอดรหัสลับอีกครั้งหนึ่งด้วยอัลกอริทึม DES จึงจะได้ข้อมูลประเภท clear text ที่แท้จริง และสามารถแสดงผลข้อมูลที่ถูกต้องได้ต่อไป



รูปที่ 4-24 แสดงถึงการใช้โปรแกรม Sniffer ในการดักจับข้อมูลประเภท cipher text โดยทำการดักจับข้อมูลจากผู้ที่ส่งข้อความดังรูปที่ 4-21

บทที่ 5
สรุป

5.1 สรุปการพัฒนาโครงงาน

จากการทำโครงงานนี้ ทำให้เราได้ศึกษาพื้นฐานการทำงานต่างๆของบัตรสมาร์ตการ์ด และได้เรียนรู้ถึงวิธีการใช้งานไมโครคอนโทรลเลอร์, หน้าจอแสดงผลแบบผลึกเหลว และการนำคีย์แพดมาใช้งานร่วมกับเครื่องอ่านสมาร์ตการ์ด เพื่อเพิ่มความปลอดภัย รวมทั้งการสร้างเครื่องอ่านเขียนสมาร์ตการ์ดขึ้นมาเพื่อประยุกต์ใช้งานกับการส่งผ่านข้อมูลต่างๆผ่านทางระบบเครือข่าย ให้มีความปลอดภัยของข้อมูลมากยิ่งขึ้น โดยใช้เทคนิคของอัลกอริทึมต่างๆในการเข้ารหัสลับข้อมูล จึงทำให้เราได้ศึกษาเรื่องราวที่เกี่ยวข้องกับกระบวนการการเข้ารหัสลับ และถอดรหัสลับของข้อมูล ซึ่งโครงงานนี้ เราจะต้องสามารถนำเอาข้อได้เปรียบในการใช้งานบัตรสมาร์ตการ์ด มาประยุกต์ใช้งานกับเทคนิคในการเข้ารหัสลับในรูปแบบต่างๆ โดยจะรวมเข้ากับทักษะในการเขียนโปรแกรมสื่อสารข้อมูลผ่านระบบเครือข่ายอีกด้วย

5.2 ปัญหาและแนวทางแก้ไข

- ในส่วนของการเขียนรหัสลับลงไปในบัตรสมาร์ตการ์ดนั้น ถ้ามีจำนวนรหัสลับมากๆ จะทำให้เกิดปัญหาคือ เครื่องอ่านเขียนสมาร์ตการ์ดรับข้อมูลที่ส่งเข้ามาไม่ทันทำให้ข้อมูลบางช่วงขาดหายไป ในบางกรณีอาจถึงกับทำให้การทำงานของเครื่องอ่านเขียนสมาร์ตการ์ดหยุดชะงักหรือค้างได้ ดังนั้นเพื่อแก้ไขปัญหานี้ จึงต้องเปลี่ยนวิธีในการส่งข้อมูลใหม่ โดยการกำหนดให้โปรแกรมวิชวลเบสิกแบ่งข้อมูลออกเป็นชุดและทำการส่งข้อมูลทีละชุด และเขียนโปรแกรมแอสเซมบลีเพิ่มเติมให้ไมโครคอนโทรเลอร์ส่งค่าตอบรับกลับไปเมื่อได้รับข้อมูลครบ 1 ชุด วิธีการเช่นนี้ให้ประสิทธิภาพที่ดีกว่าการหน่วงเวลาระหว่างการส่งข้อมูลแต่ละชุด เนื่องจากเวลาที่ใช้ในการส่งข้อมูลจริงของแต่ละชุดนั้นไม่เท่ากัน ขึ้นอยู่กับปัจจัยหลายอย่าง ถ้าเรากำหนดเวลาน้อยเกินไปก็จะทำให้ได้รับข้อมูลไม่ครบ แต่ถ้ากำหนดเวลามากเกินไปก็จะเป็นการปล่อยเวลาให้เสียไปโดยเปล่าประโยชน์
- การใช้จอแสดงผลแอลซีดีที่ต่างยี่ห้อกันทำให้เกิดปัญหาคือ ภาพที่ปรากฏขึ้นบนหน้าจอแทนที่จะเป็นตัวอักษรกลับเป็นบล็อกสี่เหลี่ยมสีดำแทน ในขั้นแรกสันนิษฐานว่าอาจจะเป็นเพราะต่อสายสัญญาณผิดหรือสายที่เชื่อมต่อระหว่างจอแสดงผลแอลซีดีกับเครื่องอ่านเขียนสมาร์ตการ์ดมีปัญหา แต่เมื่อตรวจสอบแล้วพบว่าการเชื่อมต่อถูกต้องและสายที่เชื่อมต่อก็ไม่มีปัญหาแต่อย่างใด ขั้นตอนต่อมาจึงพิจารณามาที่โปรแกรมแอสเซมบลี ซึ่งพบว่าส่วน

ของการ Initial ที่สามารถใช้ได้กับจอแสดงผลแอลซีดีอันแรก ไม่สามารถใช้ได้กับอีกอันหนึ่ง และเนื่องจากเป็นการส่งแบบ 4 บิต ดังนั้นปริมาณของ Delay ที่อยู่ระหว่างแต่ละคำสั่งจึงต้องมีมากพอที่แต่ละคำสั่งจะสามารถทำงานได้ โดยจอแสดงผลแอลซีดีย่ี่ห้อที่มีคุณภาพมากกว่าจะต้องการใช้ค่า Delay น้อยกว่า

- ปัญหาที่เกิดขึ้นจากการ Encryption และ Decryption โดยเริ่มต้นจากการรับค่าของข้อมูล ซึ่งข้อมูลนั้นอยู่ในรูปของสตริงและเนื่องจากเมื่อเราต้องการ Encryption นั้น จะต้องแปลงข้อมูลให้อยู่ในรูปของไบต์ข้อมูลเสียก่อน แล้วจึงสามารถนำไปเข้าสู่โหมดของการ Encryption ได้ แล้วจึงส่งข้อมูลที่ถูกเข้ารหัสลับ (cipher text) ออกไปยังปลายทางที่เราต้องการ แต่เนื่องจากข้อมูลดังกล่าวนั้น ถูกส่งออกไปโดยจัดให้อยู่ในฟิลด์เดียวกับข้อมูลประเภท clear text ซึ่งถูกส่งออกไปภายนอกระบบเน็ตเวิร์กและถูกจัดการให้แสดงผลที่ปลายทางในรูปของสตริง ดังนั้น ข้อมูลประเภท cipher text ดังกล่าว จึงถูกแปลงให้อยู่ในรูปของสตริง เช่นเดียวกัน และเมื่อข้อมูลถึงปลายทางแล้ว จึงค่อยนำมาแปลงกลับให้อยู่ในรูปของไบต์ข้อมูลอีกครั้งหนึ่ง เพื่อสามารถนำชุดของไบต์ข้อมูล cipher text ดังกล่าวไปเข้าสู่โหมดของการ Decryption ได้ และเราก็จะได้ข้อมูลที่ถูกถอดรหัสออกมาในรูปของไบต์ และนำมาแปลงให้อยู่ในรูปของสตริง เพื่อจัดการแสดงผลข้อมูลต่อไป แต่ขั้นตอนการทำงานดังกล่าว จะมีปัญหาเกิดขึ้น คือในบางครั้ง โปรแกรมจะเกิด exception ซึ่งเป็น Bad Padding และอีกปัญหาหนึ่งคือ บางครั้งข้อมูลที่ถูกถอดรหัสลับนั้น เราจะได้ผลลัพธ์ที่แปลกๆ ไม่เหมือนกับข้อมูลที่ยังไม่ถูกเข้ารหัสลับก่อนหน้านี้ เนื่องจากข้อมูลที่เรารับมาจากผู้ส่ง เพื่อทำการถอดรหัสลับนั้น เป็นไบต์ข้อมูลที่ถูกแปลงให้อยู่ในรูปของสตริงเสียก่อน แล้วจึงแปลงกลับมาเป็นไบต์ ไม่ใช่ไบต์ข้อมูลที่ถูกเข้ารหัสลับมาโดยตรง จึงเป็นเพียงแค่ข้อมูลที่ใกล้เคียงกับข้อมูลจริงเท่านั้น เนื่องจากรายละเอียดในอาเรย์ของไบต์ข้อมูลนั้น มีมากกว่าที่ข้อมูลประเภทสตริงจะรับได้ ดังนั้น เมื่อเราทำการแปลงไบต์ข้อมูลให้เป็นสตริง จึงมี loss เกิดขึ้น ซึ่งเป็นเหตุผลที่ทำให้ข้อมูลบางชุดที่นำมาถอดรหัสลับนั้น อาจเกิด exception หรืออาจได้ผลลัพธ์แปลกๆซึ่งอ่านไม่รู้เรื่อง ด้วยเหตุนี้ เราจึงต้องแก้ปัญหาโดยแก้ช่องทางในการสื่อสารดังกล่าว ให้จัดการส่งข้อมูลให้อยู่ในรูปของไบต์ทั้งหมด เพราะฉะนั้น เราจึงสามารถถอดรหัสลับข้อมูล ซึ่งเป็นไบต์ข้อมูลที่ถูกเข้ารหัสลับมาโดยตรง เราจึงสามารถได้ผลลัพธ์ที่ถูกต้อง หรือหากเราต้องการจะส่งข้อมูลประเภท clear text ในรูปของไบต์ข้อมูล ทางฝ่ายผู้รับก็ทำการแปลงข้อมูลกลับไปในรูปของสตริงเพื่อสามารถนำไปแสดงผลต่อไปได้

- ในการทำงานในส่วนของโปรแกรมการติดต่อ serial port ของโปรแกรม Java นั้น จะไม่สามารถรับและส่งข้อมูลพร้อมกันได้ ซึ่งในกรณีที่เราต้องการจะส่งคำสั่งกระทำการใดๆไปที่เครื่องอ่านสมาร์ตการ์ด เช่นคำสั่งในการอ่าน, เขียน หรือลบข้อมูล โปรแกรมจะเรียกใช้ object ในการ write ข้อมูลคำสั่งต่างๆดังกล่าวไปที่เครื่องอ่านสมาร์ตการ์ด จากนั้นโปรแกรมจะทำหน้าที่ในการรอรับข้อมูลตอบกลับจากเครื่องอ่านสมาร์ตการ์ด ซึ่งต้องเรียกใช้ object ในการ read โดยโปรแกรมจะต้องเช็คก่อนว่าที่ serial port นั้นมีการใช้งานหรือว่างอยู่หรือไม่ ซึ่งจากการทำงานดังกล่าว serial port ในตอนนี้กำลังถูกใช้งานโดย write object ของโปรแกรม จึงต้องมีการเขียนโปรแกรมเพื่อปิดการทำงานดังกล่าวเสียก่อน แล้วโปรแกรมจึงสามารถเข้าจองการใช้งาน serial port โดยเรียกใช้ read object ได้ เพราะฉะนั้นหากมีการส่งคำสั่งใดๆไปยังเครื่องอ่านสมาร์ตการ์ด จะต้องมีการตรวจสอบการใช้งานของ serial port ก่อนทุกครั้ง จึงจะสามารถเข้าใช้งานในเหตุการณ์นั้นๆได้

5.3 แนวทางในการพัฒนา

- การใช้สมาร์ตการ์ดชนิดโปรเซสเซอร์แทนสมาร์ตการ์ดแบบหน่วยความจำเพื่อเพิ่มความปลอดภัยให้แก่ข้อมูลมากขึ้น เนื่องจากการเข้าถึงข้อมูลจะต้องกระทำผ่านโปรเซสเซอร์ของสมาร์ตการ์ดเท่านั้น ไม่ว่าจะเป็นการอ่านหรือเขียนข้อมูลก็ตาม เพราะหน่วยความจำจะอยู่ภายในความควบคุมของโปรเซสเซอร์เพียงอย่างเดียว นอกจากนี้ยังมีการใช้ชิปประมวลผลทางคณิตศาสตร์เพื่อช่วยในการประมวลผลข้อมูลด้วยอัลกอริทึมสำหรับเข้า-ถอดรหัส ทำให้สมาร์ตการ์ดชนิดโปรเซสเซอร์มีความเร็วในการทำงานสูงกว่าสมาร์ตการ์ดชนิดหน่วยความจำหลายเท่า
- การประยุกต์ใช้ร่วมกับสมาร์ตการ์ดที่มีใช้อยู่ปกติในชีวิตประจำวัน ตัวอย่างเช่น บัตรประจำตัวพนักงานที่เป็นสมาร์ตการ์ด ซึ่งนอกจากจะใช้บัตรนี้ในการรูดบัตรเพื่อผ่านเข้า-ออกประตู และลงบันทึกเวลาทำงานแล้ว เรายังสามารถจะบันทึกรหัสลับลงไปในบัตรหรืออนำข้อมูลในบัตรที่มีอยู่มาใช้เป็นรหัสลับเพื่อใช้ในการส่งข้อมูล ซึ่งบุคคลอื่นที่ไม่ได้รับอนุญาตจะไม่สามารถอ่านข้อความดังกล่าวได้ เช่น เพื่อป้องกันบริษัทคู่แข่งดักจับข้อมูลในระหว่างการประชุมผ่านเครือข่ายของผู้บริหาร (Video Conference)
- สามารถนำกระบวนการการเข้ารหัสลับ และข้อได้เปรียบของบัตรสมาร์ตการ์ด ไปประยุกต์ใช้งานกับข้อมูลประเภทอื่นได้ เช่น ไฟล์ภาพ หรือไฟล์เสียง เป็นต้น

- การเพิ่มระดับของการป้องกัน โดยให้มีอัลกอริทึมในการเข้ารหัสลับหลากหลายรูปแบบ ตัวอย่างเช่น DES, Blowfish และอาจมีการสุ่มเลือกอัลกอริทึมขึ้นมาใช้ในทุกๆการส่งหนึ่งครั้ง นอกเหนือไปจากการสุ่มเลือกรหัสลับในบัตรสมาร์ตการ์ดเพียงอย่างเดียว
- สามารถนำไปใช้กับเครื่องมือสื่อสารหรืออุปกรณ์ทางอิเล็กทรอนิกส์อื่นๆที่ต้องการเพิ่มความปลอดภัย ตัวอย่างเช่น การนำไปใช้กับกล้องโทรทัศน์วงจรปิด

บรรณานุกรม

- ชัยวัฒน์ ลิ้มพรจิตรวิไล, วรพจน์ กรแก้ววัฒนกุล, MCS-51 Microcontroller Theory & Practical Approach: Atmel AT89C5x, บริษัท อินโนเวตีฟ เอ็กเพอริเมนต์ จำกัด
- รศ.สมยศ จุณณะปิยะ, การประยุกต์ใช้งานไมโครคอนโทรเลอร์, คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง, พิมพ์ครั้งที่ 5
- เลิศ แซ่ตั้ง, เทคโนโลยีสมาร์ตการ์ด, บริษัท ซีเอ็ดยูเคชั่น จำกัด (มหาชน)
- กิตติ ภักดีวัฒนะกุล, จำลอง ครูอุตสาหะ, Visual Basic 6 ฉบับโปรแกรมเมอร์, บริษัท เคทีพี คอมพ์แอนด์ คอนซัลท์ จำกัด, พิมพ์ครั้งที่ 8
- สัจจะ จรัสรุ่งรวีวร, คู่มือการเขียนโปรแกรมและใช้งาน Visual Basic 6, สำนักพิมพ์ อินโฟเพรส, พิมพ์ครั้งที่ 4
- อภิชาติ ภู่พลับ, เริ่มต้นเขียนโปรแกรมติดต่อและควบคุมฮาร์ดแวร์ด้วย Visual Basic, Infopress Develop Book
- อารัมภีย์ จันทร์ไข, โสรัตย์ อุนหะวรากร, เรียนรู้และเข้าใจสมาร์ตการ์ดในภาคปฏิบัติ, วารสารเซมิคอนดักเตอร์อิเล็กทรอนิกส์ ฉบับที่ 240-243 และ 246
- H.M. Deitel & P.J. Deitel, JAVA™ How to program, Prentice Hall, Fifth Edition
- ดร.วีระศักดิ์ ซึงถาวร, Java Programming Volume I,II,III, บริษัท ซีเอ็ดยูเคชั่น จำกัด (มหาชน)