หอสมุดกลาง สถาบันวิทยบริการ จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 3

การออกแบบและพัฒนาโปรแกรมเลียนแบบเทอร์มินอล

โปรแกรมเลียนแบบเทอร์มินอลภาษาไทยสำหรับเทอร์มินอล VT220 ทำงานโดยรับข้อมูลจากอุปกรณ์การสื่อสาร ด้วยการนำข้อมูลที่ได้มาแสดงผลที่จอภาพและในขณะเดียวกันโปรแกรมสามารถรับข้อมูลจากแป้นพิมพ์เพื่อส่งกลับไปให้ทางอุปกรณ์การสื่อสาร ความเร็วสำหรับการติดต่อกับอุปกรณ์สื่อสาร ผู้ใช้สามารถเลือกได้ คือ 75 300 600 1200 2400 4800 9600 19200 บิตต่อวินาที และโปรแกรมเลียนแบบเทอร์มินอลภาษาไทย สามารถแสดงผลในรูปแบบของภาษาไทยได้ โดยผู้ใช้ไม่ต้องดัดแปลงทางด้านฮาร์ดแวร์

การพัฒนาโปรแกรมเลียนแบบเทอร์มินอลภาษาไทย ได้ออกแบบและพัฒนาสำหรับเครื่องตระกูลไอบีเอ็มพีซี ซึ่งแยกการพัฒนาตามหน้าที่การทำงานออกเป็นส่วนสำคัญ ๆ ได้ 5 ส่วน คือ

3.1.1 ส่วนโปรแกรมหลัก

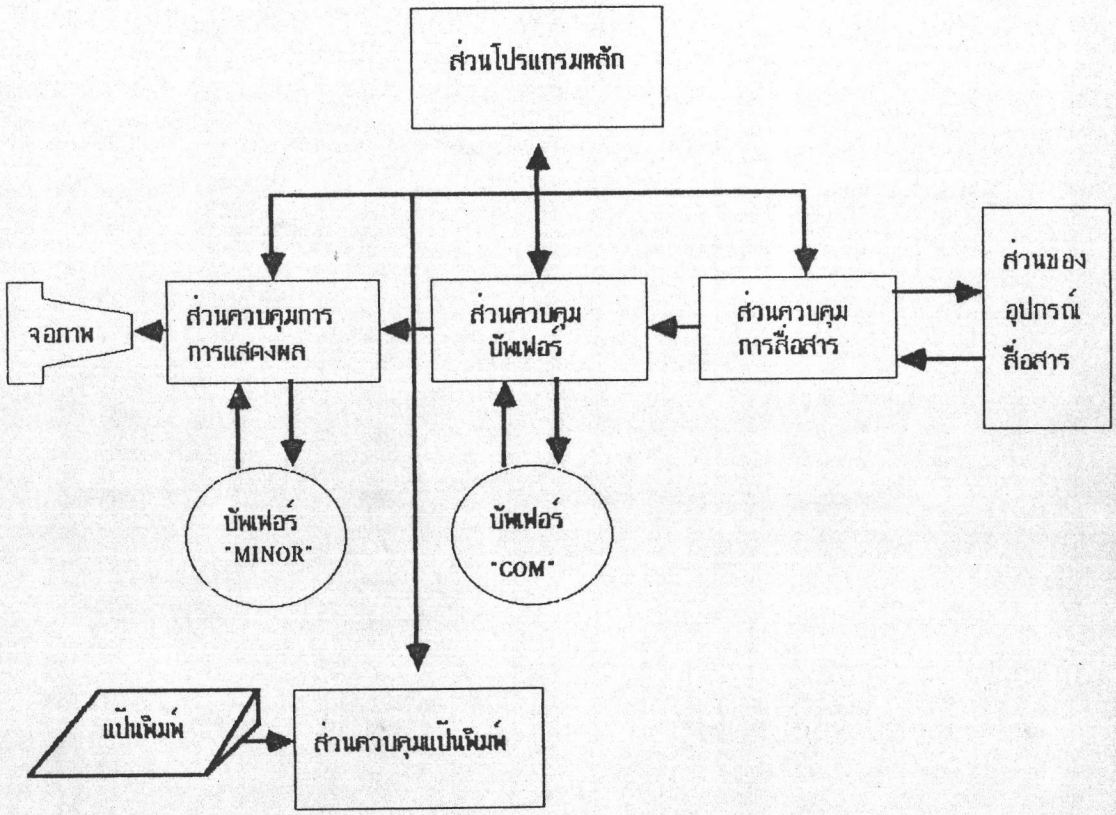
3.1.2 ส่วนควบคุมการติดต่อกับอุปกรณ์การสื่อสาร

3.1.3 ส่วนควบคุมเกี่ยวกับบัฟเฟอร์ (Buffer)

3.1.4 ส่วนควบคุมการจัดการนำข้อมูลไปแสดงผลที่จอภาพ

3.1.5 ส่วนควบคุมเกี่ยวกับแป้นพิมพ์

และมีผังงานการทำงานของแต่ละส่วนดังรูปที่ 3.1 การทำงานของแต่ละส่วนสามารถอธิบายได้ดังต่อไปนี้



รูปที่ 3.1 แสดงผังการทำงานของส่วนควบคุมแต่ละส่วน

3.1.1 ส่วนโปรแกรมหลัก

ส่วนโปรแกรมหลักเป็นส่วนที่ควบคุมการทำงานของส่วนควบคุมอื่น ๆ โดยทำหน้าที่เป็นตัวเชื่อมการทำงานของแต่ละส่วนเข้าด้วยกัน จากผังงานการทำงานในรูปที่ 3.1 โปรแกรมหลักมีหน้าที่เริ่มต้นส่วนควบคุมการสื่อสาร ซึ่งจะกล่าวต่อไป ถ้ามีข้อมูลเข้ามาจากอุปกรณ์ภายนอก ส่วนควบคุมการสื่อสารจะรับข้อมูลแล้วส่งไปให้กับส่วนควบคุมบัฟเฟอร์ ส่วนควบคุมบัฟเฟอร์ก็จะนำข้อมูลมาใส่ในบัฟเฟอร์ "COM" และส่วนโปรแกรมหลักจะตรวจว่ามีข้อมูลในบัฟเฟอร์หรือไม่ ถ้ามีส่วนของโปรแกรมหลักก็จะนำข้อมูลจากบัฟเฟอร์ "COM" ไปส่งให้กับส่วนควบคุมการจัดนำข้อมูลไปแสดงผลที่จอภาพเพื่อแสดงผลที่จอภาพต่อไป ในกรณีที่บัฟเฟอร์เต็มเนื่องจาก ส่วนควบคุมการนำข้อมูลไปแสดงผลที่จอภาพ ไม่ทันกับการนำข้อมูลจากอุปกรณ์สื่อสารมาใส่ในบัฟเฟอร์ "COM" ส่วนโปรแกรมหลักก็จะส่ง XOFF ผ่านทางส่วนควบคุมการสื่อสาร เพื่อไปหยุดการส่งข้อมูลจากคอมพิวเตอร์แม่ จนกระทั่งบัฟเฟอร์ "COM" มีที่ว่างพอสำหรับข้อมูลชุดต่อไป โปรแกรมหลักก็จะส่งXONผ่านทางส่วนควบคุมการสื่อสาร เพื่อให้คอมพิวเตอร์แม่ส่งข้อมูลชุดต่อมาได้อีก สำหรับส่วนควบคุมแป้นพิมพ์ในส่วนโปรแกรมหลักจะตรวจสอบว่ามีข้อมูลอยู่หรือไม่ ถ้ามีก็ส่งไปให้กับส่วนควบคุมการสื่อสารทันที เพื่อที่จะส่งให้อุปกรณ์การสื่อสารอีกที

3.1.2 ส่วนควบคุมการจัดการติดต่อกับอุปกรณ์การสื่อสาร

การนำข้อมูลเข้าและออกระหว่างอุปกรณ์สื่อสาร และเครื่องไอบีเอ็มพีซีใช้หลักการของการอินเทอร์รัพ โดยมีไอซี 8250 เป็นตัวควบคุมเกี่ยวกับการรับและส่งข้อมูล ซึ่งจะกล่าวต่อไป ส่วนควบคุมการสื่อสาร เป็นส่วนกำหนดค่าเริ่มต้นของอุปกรณ์การสื่อสาร กำหนดเกี่ยวกับบอทเรต สต็อปบิต (Stop Bit) พาริตี้บิต (Parity Bit) และกำหนดตำแหน่งเริ่มต้นของรูทีนที่ทำงาน เมื่อมีอินเทอร์รัพของอุปกรณ์สื่อสารเกิดขึ้น การกำหนดตำแหน่งเริ่มต้นของรูทีนทำได้ โดยไปแก้ไขตารางอินเทอร์รัพของเครื่องไอบีเอ็มพีซีที่ตำแหน่ง 12 (COM1)หรือตำแหน่งที่ 11(COM2) ให้ชี้ไปที่ตำแหน่ง

เริ่มต้นการทำงานของรูทีน "AsynInterrupt" โดยรูทีนทำหน้าที่นำเอาข้อมูลจากอุปกรณ์สื่อสารส่งไปให้กับส่วนควบคุมบัฟเฟอร์เพื่อจะนำข้อมูลไปไว้ในบัฟเฟอร์

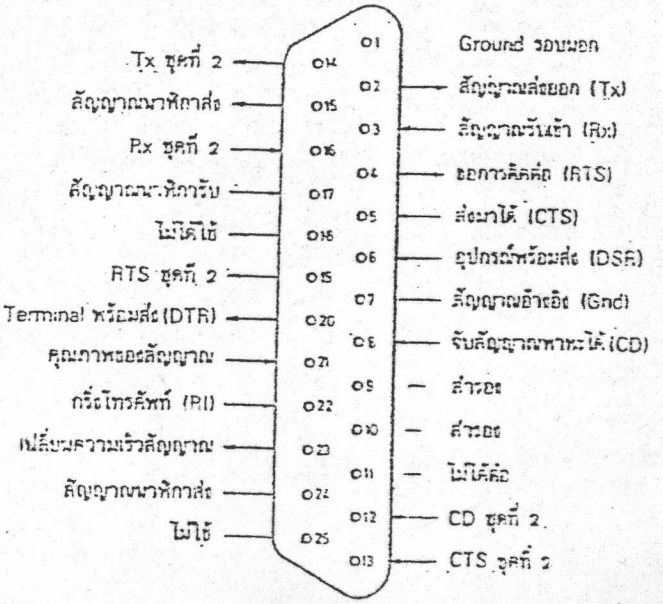
อุปกรณ์การสื่อสารที่ไอบีเอ็มพีซีใช้มาตรฐานอาร์เอส232 ซี (RS 232C) ตามมาตรฐานของสมาคมอุตสาหกรรมอิเลกทรอนิกส์ (EIA) แต่อาร์เอส 232ซี ที่เครื่องไอบีเอ็มพีซีใช้จะแตกต่างกับมาตรฐานตรงที่ สัญญาณที่ส่งออกแทนที่จะเป็น +15 โวลท์ และ -15 โวลท์ กลับเป็นแค่ 5 โวลท์ และ 0 โวลท์ และเพิ่มการรับส่งแบบกระแสวนกลับ 20 มิลลิแอมป์ ข้อดีของการส่งแบบกระแสวนกลับ ก็ตรงที่ระยะทางของการส่งจะไปได้ไกลกว่าแรงดันไฟฟ้าของอาร์เอส 232 ซี ที่กำหนดโดยสมาคมอุตสาหกรรมอิเล็กทรอนิกส์

หัวใจสำคัญของอุปกรณ์การสื่อสารที่ใช้มาตรฐานของอาร์เอส232 ซี ก็คือไอซี8250 โดยสามารถกำหนดบอทเรต ได้ภายในตัวและมีรีจิสเตอร์ที่ใช้ควบคุม 7 รีจิสเตอร์ด้วยกันซึ่งจะกล่าวต่อไป นอกเหนือจากนั้นเป็นพวกไอซีที่ทำหน้าที่แยกตำแหน่งที่อยู่ของช่องทางเข้าออก เครื่องไอบีเอ็มพีซีสำรองช่องการสื่อสารแบบอนุกรมไว้ 2 ช่องทาง โดยดอส(DOS) จะรับรู้ว่ามีช่องทางการสื่อสารได้เพียง 2 ช่องทางเท่านั้น แต่ไม่ได้หมายความว่า เราจะทำให้การสื่อสารแบบอนุกรมมากกว่าสองไม่ได้ เพียงแต่ว่าเราต้องสร้างโปรแกรมควบคุมเอง

เครื่องไอบีเอ็มพีซีกำหนดหมายเลขของช่องทางเข้าออกสำหรับรีจิสเตอร์ควบคุมของ 8250 และการใช้งาน ดังในตารางที่ 3.1 และช่องทางการสื่อสารของเครื่องไอบีเอ็มพีซีใช้หมายเลขช่องทางที่เริ่มต้นด้วย 3Fx สำหรับช่องทางการสื่อสารที่ 1 (COM1) และ 2Fx สำหรับช่องทางการสื่อสารที่ 2 (COM2)

สำหรับหมายเลขของทางเข้าออกที่ 3F8 หรือ 2F8 และ 3F9 หรือ 2F9 นั้นใช้สำหรับ 2 หน้าที่ โดยจะรวมกันเป็นตัวหาร (DLM และ DLL) ถ้าบิตที่ 8

ของ IER เป็นหนึ่ง8250 ทำการสุ่มตัวอย่าง สัญญาณที่รับเข้ามาเป็น 16 เท่า ของอัตราการรับส่งข้อมูลวงจรออสซิลเลเตอร์ในอะแดปเตอร์เองทำความถี่ 1.8432



รูปที่ 3.2 กำหนดมาตรฐานของสัญญาณการติดต่อของอาร์เอส 232 ซี

เมกกะเฮิรซ์ ป้อนให้กับ 8250 การกำหนดตัวหารเพื่อให้เกิดเป็นบอทเรตที่ต้องการแสดงอยู่ในตารางที่3.2 ดังนั้นส่วนควบคุมเกี่ยวกับบัพเฟอร์ ส่งข้อมูลไปให้กับ 8250 เพื่อส่งไปให้กับอุปกรณ์สื่อสาร สามารถส่งผ่านไปยังตำแหน่ง 3F8 สำหรับช่องทางการสื่อสารช่องที่ 1 และ 2F8 สำหรับช่องทางการสื่อสารที่ 2 และส่วนควบคุมบัพเฟอร์สามารถรับข้อมูลจากอุปกรณ์สื่อสาร โดยผ่านตำแหน่งการติดต่อเช่นเดียวกันกับการส่งข้อมูล แต่ถ้าบิตที่ 8 เป็นศูนย์ 2F8 หรือ 3F8 จะทำหน้าที่เป็นที่พักของข้อมูลทั้งรับและส่ง (2 รีจิสเตอร์ใช้หมายเลขของช่องทางออกอันเดียวกัน) ส่วน 2F9 หรือ 3F9 จะใช้สำหรับควบคุมการเกิดอินเทอร์รัพ โดยมีความหมายของการควบคุมดังแสดงในตารางที่ 3.3

ช่องการสื่อสารที่ 1	ช่องการสื่อสารที่ 2	รีจิสเตอร์	การใช้งาน
3F8	2F8	RBR DLL	-ใช้สำหรับที่พักข้อมูลทั้งรับและส่ง -ใช้เป็นตัวหารตัวที่มีค่าน้อยสำหรับหารความถี่ 1.8432 MHzเป็น Baud Rate
3F9	2F9	DLM IER	-ใช้เป็นตัวหารที่มีค่าสูงสำหรับหารความถี่ 1.8432 MHZ ให้เป็น Baud Rate -ใช้สำหรับควบคุมการเกิดอินเทอร์รัพต์
3FA	2FA	IIR	-สำหรับบอกชนิดของอินเทอร์รัพต์
3FB	2FB	LCR	-สำหรับควบคุมพารามิเตอร์การรับส่ง -สำหรับการบ่งบอกการใช้งานของ DLL และ DLM
3FC	2FC	MCR	-ใช้สำหรับควบคุมโมเด็ม
3FD	2FD	LSR	-สำหรับบอกสถานภาพของการรับส่ง
3FE	2FE	MSR	-สำหรับบอกสถานภาพของโมเด็ม

ตารางที่ 3.1 แสดงช่องทางเข้าออกและการใช้งานของรีจิสเตอร์ใน 8250

สำหรับรีจิสเตอร์ IIR ที่หมายเลขทางเข้าออก 3FA หรือ 2FA ใช้สำหรับบอกว่าอินเทอร์รัพชนิดไหน ใน 4 ชนิด ที่เราสามารถกำหนดให้เกิดได้โดยรีจิสเตอร์ IER ในโปรแกรม สำหรับจัดการอินเทอร์รัพ จะต้องมาตรวจสอบรีจิสเตอร์ตัวนี้ก่อนเพื่อที่จะได้จัดการได้ถูกต้องว่าเกิดอินเทอร์รัพเพราะเหตุใด การอ่านค่าจากหมายเลขช่องทางเข้าออกนี้จะบอก ได้โดยบิตที่ 0 จะเป็นตัวบอกว่าสัญญาณอินเทอร์รัพยังอยู่หรือหายไป (0 ยังอยู่ 1 หายไป) ถ้าหากเกิดมีอินเทอร์รัพหลายชนิดซ้อนกัน ชนิดที่มีความสำคัญที่สุดจะปรากฏในรีจิสเตอร์ดังตารางที่ 3.4 บอกชนิดอินเทอร์รัพที่เกิด พร้อมทั้งการกระทำที่จะทำให้สัญญาณอินเทอร์รัพหายไป

Baud	ตัวหาร	
	ฐาน 10	ฐาน 16
300	384	180
600	192	0C0
1200	96	060
2400	48	030
4800	24	018
9600	12	00C
19200	6	006

ตารางที่ 3.2 ตัวหาร 1.843 MHz เพื่อให้ได้ความถี่ 16 เท่าของบอทเรต

บิตที่	0	สร้างสัญญาณอินเตอร์รัพต์เมื่อรับข้อมูลได้ครบ 1 อักขระ
บิตที่	1	สร้างสัญญาณอินเตอร์รัพต์เมื่อรีจิสเตอร์สำหรับส่งว่างลง
บิตที่	2	สร้างสัญญาณอินเตอร์รัพต์เมื่อมีข้อพลาดเกิดขึ้นจากการรับ
บิตที่	3	สร้างสัญญาณอินเตอร์รัพต์เมื่อมีสัญญาณ Modem Status

ตารางที่ 3.3 ความหมายของบิตการควบคุมการกำเนิดสัญญานอินเตอร์รัพ

ลำดับความสำคัญ	ค่าที่อ่านได้ใน IIR	สาเหตุของการอินเตอร์รัพท์	ขบวนการทำให้สัญญาณอินเตอร์รัพต์หาย
1	0000110	-เกิดการเขียนทับข้อมูลเก่า -ตรวจสอบผิดพลาด -ผิดพลาดของช่วงอักขระ	อ่านรีจิสเตอร์ LSR ที่หมายเลขช่องทาง XFD
2	0000100	-เมื่อรับข้อมูลครบอักขระ	อ่านข้อมูลจาก RBR ที่หมายเลขช่องทาง XF8
3	0000010	-เมื่อส่งข้อมูลออกไปหมด	เขียนข้อมูลลง RBR

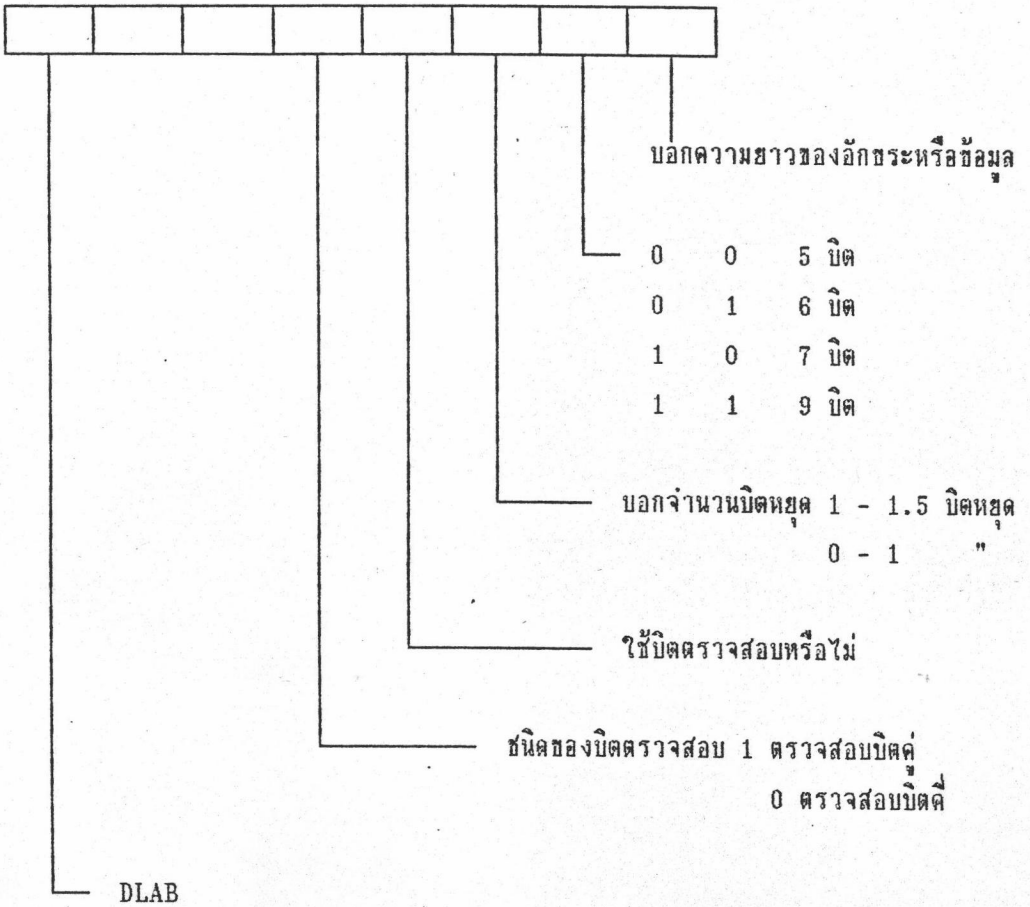
ตารางที่ 3.4 รีจิสเตอร์ IIR แสดงชนิดของอินเตอร์รัพที่เกิดขึ้น

การควบคุมพารามิเตอร์ของการรับส่ง สามารถทำได้โดยการกำหนดค่าลงในรีจิสเตอร์ LCR โดยค่าที่กำหนดนี้มาจากส่วนภาวะเซ็ตอัพ (Set Up State) ที่ผู้ใช้เป็นผู้เลือกที่หมายเลขช่องทาง 2FB หรือ 3FB โดยค่าใน 2FB หรือ 3FB มีความหมายดังรูปที่ 3.3

ปัญหาที่เกิดขึ้นสำหรับส่วนควบคุมการสื่อสารก็คือ การเกิดการรับข้อมูลไม่ทัน(overrun) เนื่องจากส่วนควบคุมการสื่อสารไม่สามารถนำข้อมูลจากอุปกรณ์สื่อสารไปส่งให้กับส่วนควบคุมบัพเฟอร์ เพื่อนำไปเก็บในบัพเฟอร์และกลับมารับข้อมูลจากอุปกรณ์การสื่อสารได้ทัน เมื่อมีข้อมูลตัวใหม่เข้ามา ทางแก้ปัญหาก็คือต้องพยายามลดช่วงเวลาในการนำข้อมูลจากอุปกรณ์การสื่อสาร ไปส่งให้กับส่วนควบคุมบัพเฟอร์ให้น้อยที่สุด และหลีกเลี่ยงการใช้อินเทอร์รัพที่มีระดับความสำคัญสูงกว่าอินเทอร์รัพของอุปกรณ์

017663

การสื่อสารอย่างเช่นการเข้าไปแก้ไขอินเทอร์รัพของเวลา (Timer Interrupt) เพื่อให้แสดงเวลาที่หน้าจอเสมอ



รูปที่ 3.3 การตั้งค่าพารามิเตอร์ของการสื่อสารที่หมายเลขช่องทางเข้าออก 2FB หรือ 3FB

3.1.3 ส่วนควบคุมเกี่ยวกับบัพเฟอร์ (Buffer Control Routine)

ในส่วนควบคุมบัพเฟอร์นี้ออกแบบเพื่อจัดการเกี่ยวกับการนำข้อมูลมาใส่ในบัพเฟอร์และนำข้อมูลออกจากบัพเฟอร์ โดยลักษณะของบัพเฟอร์ที่ใช้จะเป็นแบบการเรียงต่อเป็นแบบวงกลม (Circular queue) ทำให้ประหยัดเนื้อที่ของหน่วยความจำที่

ใช้สำหรับเก็บข้อมูลรูทีนที่ทำงาน ในส่วนควบคุมเกี่ยวกับบัพเฟอร์เกี่ยวข้องกับตัวแปรทั้งหมด 3 ตัวแปรคือ

1. fronta ทำหน้าที่ชี้ตำแหน่งของข้อมูลที่ต้องการนำออกมาจากบัพเฟอร์

2. reara ทำหน้าที่ชี้ตำแหน่งของข้อมูลที่ต้องการนำข้อมูลมาเก็บในบัพเฟอร์

3. counta ทำหน้าที่บอกจำนวนตัวอักษรที่อยู่ในบัพเฟอร์ และในส่วนโปรแกรมหลักจะตรวจสอบตัวแปรตัวนี้ว่าขณะนี้บัพเฟอร์มีขนาดเท่าไร

รูทีนที่เกี่ยวกับการทำงานของบัพเฟอร์มีอยู่ 3 รูทีน คือ

3.1.3.1 nexta

จุดประสงค์ในการออกแบบรูทีนนี้เพื่อนำข้อมูลในบัพเฟอร์ออกมา การทำงานของรูทีนนี้จะตรวจสอบจำนวนข้อมูลในบัพเฟอร์ว่ามีข้อมูลหรือไม่ ถ้ามีก็จะนำข้อมูลออกมา และเปลี่ยนแปลงค่า fronta โดย

fronta = (fronta + 1) modulo ขนาดของบัพเฟอร์

และลดค่า counta ลง 1 ค่า ถ้าไม่มีข้อมูลอยู่ก็จะส่งค่า "\0" กลับไป

3.1.3.2 puta

จุดประสงค์ในการออกแบบรูทีนนี้เพื่อนำข้อมูลไปเก็บไว้ในบัพเฟอร์ ถ้าบัพเฟอร์เต็มก็จะกลับมาที่โปรแกรมหลักทันที ถ้าบัพเฟอร์ไม่เต็มก็จะนำข้อมูลใส่ในบัพเฟอร์และเปลี่ยนแปลงค่า reara โดย

reara = (reara + 1) modulo ขนาดของบัพเฟอร์

และเพิ่มค่า counta อีก 1 ค่า

3.1.3.3 empta

จุดประสงค์ในการออกแบบรูทีนนี้เพื่อตรวจสอบว่าในบัพเฟอร์มีข้อมูลอยู่หรือไม่ ถ้ามีก็จะส่งค่า "0" ถ้าไม่มีก็ส่งค่า "1" ออกมาให้กลับโปรแกรมหลัก

3.1.4 ส่วนควบคุมการจัดการนำข้อมูลไปแสดงผลที่จอ

การออกแบบโปรแกรมในส่วนนี้เป็นส่วนที่ยุ่งยากและซับซ้อน เนื่องจากการแสดงผลอยู่ในโหมดกราฟฟิค และสามารถแสดงผลได้ทั้งการ์ดเฮอร์คิวลีส อีจีเอ และวีจีเอ โดยผู้ใช้ไม่ต้องคอยตรวจเช็คว่าการ์ดที่ใช้เป็นชนิดไหน โปรแกรมเป็นตัวตรวจสอบให้หมด สำหรับการ์ดต่าง ๆ เหล่านี้มีคุณสมบัติเฉพาะของตัวเอง ดังนั้นการออกแบบการแสดงผลต้องครอบคลุมการทำงานของทั้ง 3 การ์ดนี้ทั้งหมด

การตรวจสอบชนิดของการ์ดว่าเป็นชนิดไหน ได้ใช้ฟังก์ชั่นของเทอร์โบซี (Turbo C) เป็นตัวตรวจสอบ โดยฟังก์ชั่นมีชื่อว่า

```
detectgraph();
```

หลังจากได้ชนิดของการ์ดแล้วจะนำไดรเวอร์ที่เหมาะสมสำหรับการ์ดนั้นๆลง โดยใช้ฟังก์ชั่น

```
initgraph();
```

โดยไดรเวอร์ที่นำมาลงถูกเก็บเป็นแฟ้มอยู่ 2 แฟ้มคือ

1. herc.bgi เป็นไดรเวอร์สำหรับการ์ดเฮอร์คิวลีส
2. egavga.bgi เป็นไดรเวอร์สำหรับการ์ดวีจีเอกับการ์ดอีจีเอ

ในส่วนควบคุมการทำงานของการนำข้อมูลไปแสดงผลที่จอ ได้แบ่งการทำงานตามจุดประสงค์ออกเป็น 3 ส่วน คือ

3.1.4.1 <u>ส่วนควบคุมเกี่ยวกับแอดเดรสของการ์ดแสดงผล</u>

ลักษณะการควบคุมการแสดงผลของการ์ดทั้ง 3 ชนิดนี้ควบคุมที่ตำแหน่งการแสดงผล สำหรับการ์ดเฮอร์คิวลีสมีรีโซลูชั่น(Resolusion) ขนาด 720 x 348 ตำแหน่งเริ่มต้นการแสดงผลของการ์ดเฮอร์คิวลีสอยู่ที่ตำแหน่ง 0xB000:0000 ของเพจ 0 และ 0xB800:0000 ของเพจ 1 ส่วนลักษณะของการ์ดวีจีเอ และ อีจีเอ มีรีโซลูชั่น ขนาด 640 x 350 ตำแหน่งเริ่มต้นการแสดงผลอยู่ที่ตำแหน่ง 0xA000:0000 ของเพจ 0 และ 0xA800:0000 ของเพจ 1

การแสดงผลทำได้โดยการนำเอาข้อมูลไปใส่ในตำแหน่งข้อมูลที่เหมาะสมของการ์ดแสดงผลนั้น ๆ โดยการแสดงผลบนจอภาพกำหนดเป็นจุด (Pixel) และแต่ละจุดบนจอภาพมีความสัมพันธ์กับตำแหน่งบนหน่วยความจำของการ์ดแสดงผลนั้น โดยการ์ด

เฮอร์คิวลีสมีการอ้างตำแหน่งจุดตามแนวแกน (x,y) ที่มีความสัมพันธ์กับหน่วยความจำดังนี้คือ

ตำแหน่งหน่วยความจำ = ((y & 3) << 13) + 90 * (y >> 2) + (x >> 3)

สำหรับการ์ดอีจีเอ วีจีเอ มีการอ้างตำแหน่งจุดตามแนวแกน (x,y) ที่มีความสัมพันธ์กับหน่วยความจำดังนี้คือ

ตำแหน่งหน่วยความจำ = 80 * (y >> 2) + (x >> 3)

ขนาดของตัวอักษรที่ใช้ในการแสดงผลคือขนาดความกว้าง8จุดและ ความยาวขนาด 14 จุด

การอ้างตำแหน่งของจุดบนจอภาพ จะไม่ใช้การคำนวณว่าจุดที่แสดงผลอยู่ที่ตำแหน่งแสดงผลที่เท่าไรตามสูตรที่ให้ไว้ข้างบน แต่จะใช้การเก็บค่าที่แน่นอนของตำแหน่งหน่วยความจำที่มีความสัมพันธ์กับจุดเริ่มต้นการแสดงผล เช่นจุดที่ 0 ของบรรทัดที่ 0 ของการ์ดเฮอร์คิวลีส จะมีตำแหน่งแสดงผลที่ 0xB000:0000 และจุดที่ 0 ของบรรทัดที่ 1 ของการ์ดเฮอร์คิวลีส จะมีตำแหน่งแสดงผลที่ 0xB000:2000 ในการเก็บตำแหน่งหน่วยความจำ เก็บตำแหน่งหน่วยความจำเริ่มต้นของแต่ละบรรทัดเท่านั้น และ ตำแหน่งหน่วยความจำของการ์ดอีจีเอ กับวีจีเอ จะเหมือนกันดังนั้นการเก็บตำแหน่งหน่วยความจำเก็บเฉพาะการ์ดเฮอร์คิวลีสกับอีจีเอ ส่วนการนำค่ามาใส่ในแต่ละจุดบนจอภาพ จะนำเพียงค่าออฟเซ็ต (offset) มาบวกกับตำแหน่งเริ่มต้นของแต่ละบรรทัดเท่านั้น

ข้อดีของการเก็บตำแหน่งของหน่วยความจำเริ่มต้นของแต่ละการ์ด ไว้คือ ทำให้ประะหยัดเวลาในการอ้างตำแหน่งของจุดบนจอภาพ เนื่องจากการอ้างถึง

ตำแหน่งของจุดบนจอภาพที่มีความสัมพันธ์กับตำแหน่งบนหน่วยความจำ ถ้าใช้การคำนวณจะใช้เวลามากแต่ข้อเสียคือ ต้องใช้หน่วยความจำสำหรับเก็บค่าเริ่มต้นของหน่วยความจำของการ์ดแต่ละชนิดเพิ่มขึ้น และขนาดของโปรแกรมก็มีขนาดใหญ่ขึ้น

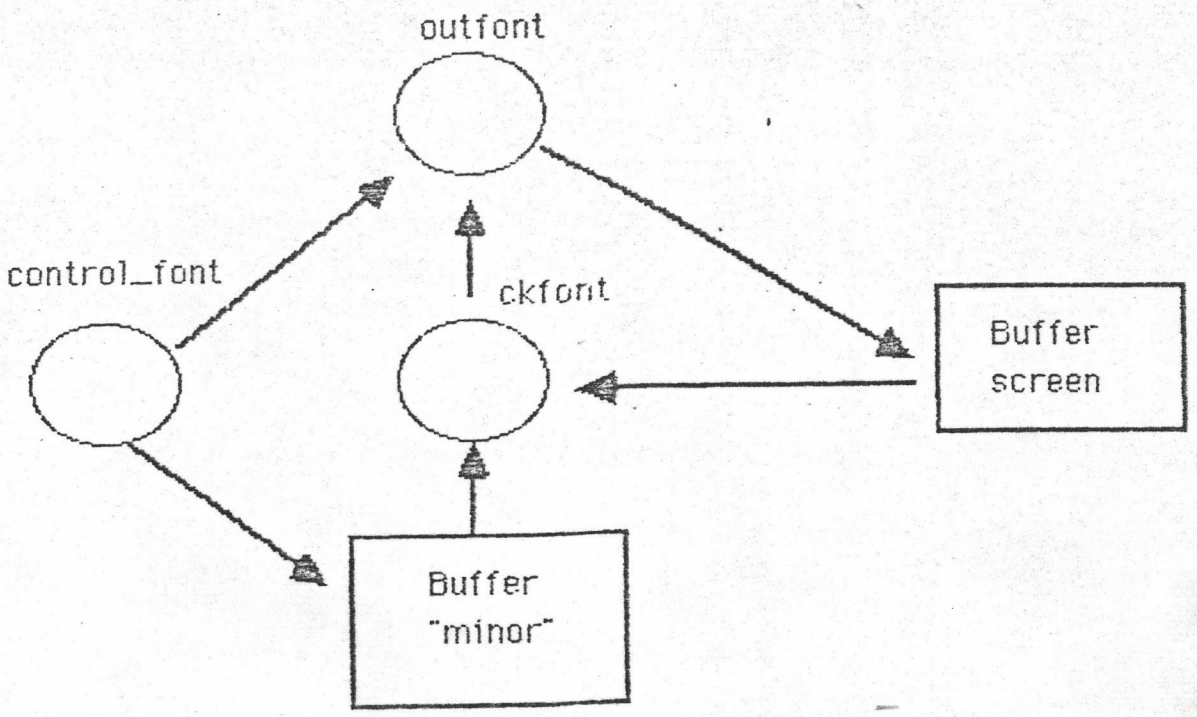
รูทีนที่ใช้ ในส่วนควบคุมเกี่ยวกับแอดเดรสของการ์ดแสดงผลก็คือรูทีน "outfont" รูทีนนี้ทำหน้าที่รับค่าตัวอักษรและตำแหน่งที่ต้องการแสดงผลบนจอภาพ โดยทำการเปลี่ยนแปลงตัวอักษรที่รับมาให้เป็นค่าที่เหมาะสมสำหรับการแสดงผลแล้วจึงหาตำแหน่งหน่วยความจำที่เหมาะสมของจอภาพ เพื่อนำค่าที่เหมาะสมสำหรับการแสดงผลไปใส่

3.1.4.2 <u>ส่วนควบคุมเกี่ยวกับอักขระพิเศษ</u> (Escape Sequence Control)

ส่วนควบคุมเกี่ยวกับอักขระพิเศษ ทำหน้าที่ตรวจสอบอักขระพิเศษ ที่ส่งมาทางช่องทางสื่อสาร เพื่อที่จะให้เทอร์มินอลทำหน้าที่พิเศษบางอย่างตามหน้าที่ที่ถูกกำหนดไว้ สำหรับเทอร์มินอลชนิดนั้น สำหรับเทอร์มินอล VT220ลักษณะของอักขระพิเศษจะนำหน้าด้วยระหัส 27(เลขฐาน 10)หรือ escape character (ESC) และตามด้วยอักขระจำนวนหนึ่งเพื่อกำหนดหน้าที่การทำงานของอักขระพิเศษชุดนั้น ๆ

3.1.4.3 <u>ส่วนควบคุมเกี่ยวกับภาษาไทย</u> (Thai Management)

รูทีนควบคุมภาษาไทยควบคุมเกี่ยวกับการจัดตำแหน่งของสระ และวรรณยุกต์และการเลื่อนตำแหน่งเคอร์เซอร์บนจอภาพ การไหลของข้อมูลสามารถแสดงได้ดังรูปที่ 3.4



รูป 3.4 แสดงการไหลของข้อมูลของส่วนควบคุมภาษาไทย

จากรูป 3.4 การทำงานของส่วนนี้แยกบัฟเฟอร์ออกเป็น 2 ส่วนคือ

1. บัฟเฟอร์สำหรับตัวอักษร(Buffer"minor") ซึ่งถูกควบคุมโดยรูทีน"control_font" ลักษณะของข้อมูลที่เก็บ มีตำแหน่งสัมพันธ์กับตำแหน่งบนจอภาพ

2. บัฟเฟอร์สำหรับจอภาพ (Buffer Screen) ซึ่งถูกควบคุมโดยรูทีน"outfont"

ลักษณะการทำงานส่วนของโปรแกรมหลักตรวจสอบว่า มีข้อมูลในบัพเฟอร์ "COM" หรือไม่ โดยใช้รูทีน "empta" ถ้ามีก็จะนำข้อมูลจากบัพเฟอร์ โดยใช้รูทีน "nexta" จากนั้น ก็ส่งข้อมูลให้กับรูทีน "control_font" โดยรูทีน "control_font" มีหน้าที่นำข้อมูลไปเก็บไว้ในบัพเฟอร์ "minor" ในตำแหน่งที่ถูกต้อง และจะส่งตัวอักษรและตำแหน่งของตัวอักษรที่ต้องแสดงที่หน้าจอภาพ ไปให้รูทีน "outfont" เพื่อแสดงผลบนจอภาพ หลังจากนั้นรูทีน"ckfont" ตรวจสอบตำแหน่งของข้อมูลถัดไปว่าถูกต้องหรือไม่ ถ้าไม่ก็จะนำเอาข้อมูลที่ถูกต้องจากบัพเฟอร์ "minor" ส่งให้กับรูทีน "outfont" เพื่อแสดงผลให้ถูกต้อง

3.1.5 ส่วนควบคุมเกี่ยวกับแป้นพิมพ์

ก่อนอธิบายเกี่ยวกับส่วนควบคุมเกี่ยวกับแป้นพิมพ์ จะกล่าวถึงลักษณะการทำงานของแป้นพิมพ์ของเครื่องพิมพ์ไอบีเอ็มพีซีก่อน แป้นพิมพ์มีบัพเฟอร์ของแป้นพิมพ์เอง ซึ่งสามารถรับตัวอักษรได้ 15 ตัวอักษร การที่นำข้อมูลออกจากบัพเฟอร์ของแป้นพิมพ์ได้นั้น ใช้โปรแกรมอินเทอร์รัพที่ไบออส (BIOS) ได้จัดเตรียมไว้ให้ โดยโปรแกรมอินเทอร์รัพที่ไบออสจัดเตรียมไว้ให้มีหลายโปรแกรมขึ้นอยู่กับลักษณะงานที่ผู้ใช้ต้องการ เนื่องจากโปรแกรมเลียนแบบเทอร์มินอลภาษาไทย ได้ใช้หลักการของกราฟฟิก การหยุดการทำงานของโปรแกรมโดยการกด ctrl cต้องไม่สามารถทำได้ ดังนั้นส่วนควบคุมแป้นพิมพ์ต้องสามารถตรวจสอบการกด ctrl c ได้ การแก้ปัญหาในส่วนนี้ใช้โปรแกรมอินเทอร์รัพหมายเลข 16 ฟังก์ชั่น 1 และ 0 ไว้คอยตรวจสอบค่าที่ได้จากการกดคีย์ ถ้าเป็นการกด ctrl c ค่าที่ได้จากโปรแกรมอินเทอร์รัพหมายเลข 16 คือค่า 0x03

จุดประสงค์ของส่วนควบคุมแป้นพิมพ์นี้ เพื่อรับค่าจากแป้นพิมพ์แล้วส่งให้กับอุปกรณ์สื่อสาร การทำงานของส่วนควบคุมแป้นพิมพ์แบ่งเป็น 2 ส่วนคือ

3.1.5.1 การทำงานแบบภายใน (Local Mode)

การทำงานภายในคือ การทำงานโดยไม่ส่งค่าไปให้กับอุปกรณ์สื่อสาร โดยการทำงานภายในจะเกิดขึ้นได้จากการกดคีย์ 2 แบบ คือ

3.1.5.1.1 การกดคีย์ Alt และ s พร้อมกันทันที

ถ้ามีการกดคีย์นี้รูทีนเปลี่ยนไปทำฟังก์ชั่นเกี่ยวกับการเซ็ตอัพระบบและจะรอจนกระทั่งมีการกดคีย์นี้พร้อมกันอีกที เพื่อกลับไปทำงานเดิม

3.1.5.1.2 การกดคีย์ Alt และ t พร้อมกันทันที

ถ้ามีการกดคีย์นี้รูทีนจะทำการสลับโหมดของตัวอักษรจากไทยเป็นอังกฤษ หรือ จากอังกฤษเป็นไทย โดยเปลี่ยนรูปแบบภาษาไทยตามที่กำหนดไว้ในโหมดเซ็ตอัพ (เป็น ระหัส ส.ม.อ. หรือ เกษตร) โดยการทำงานแบบนี้จะเปลี่ยนค่าของตัวแปร thaiflag โดยถ้า

thaiflag เป็น 0 ค่าที่รับจากแป้นพิมพ์เป็นภาษาอังกฤษ
thaiflag เป็น 1 ค่าที่รับจากแป้นพิมพ์เป็นภาษาไทย

ในกรณีที่ thaiflag เป็น 1 ค่าที่รับจากแป้นพิมพ์จะถูกเปลี่ยนเป็นค่าภาษาไทยที่เหมาะสม เช่นที่แป้นพิมพ์ "a" จะตรงกับ "ฟ" ในภาษาไทย ค่ารหัสของ "a" ต้องถูกเปลี่ยนเป็นค่าของรหัส "ฟ" ก่อนส่งให้กับอุปกรณ์สื่อสาร

3.1.5.2 การส่งผ่านค่า

ส่วนควบคุมแป้นพิมพ์จะส่งค่าที่ได้จากแป้นพิมพ์ไปให้อุปกรณ์สื่อสารทันที ก็ต่อเมื่อค่านั้นไม่ใช่ค่าสำหรับการทำงานภายในคือการกดค่าคีย์ Alt s และ Alt t เท่านั้น