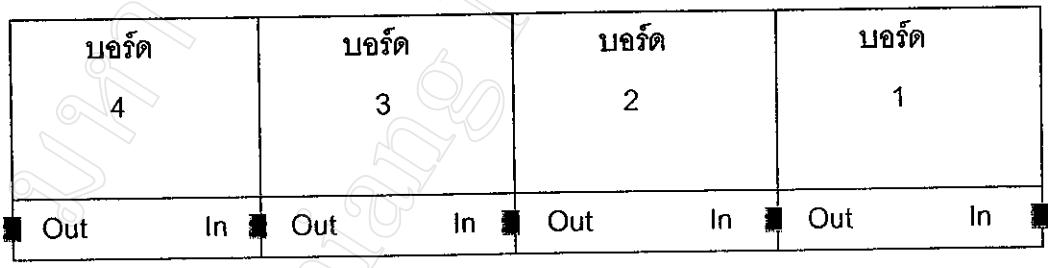
บทที่ 2
ลักษณะโครงสร้างและการทำงานของแผงไฟอักษรวิ่ง

ความสำคัญของบทนี้กล่าวถึงโครงสร้างภายนอกของแผงไฟ และวิธีการใช้ พอร์ทเข้าออก (สัญญาณเข้าออก พอร์ท) เพื่อควบคุมสัญญาณต่าง ๆ รวมไปถึงวิธีการส่งข้อมูลให้กับระบบแผงไฟ และวิธีการจัดเก็บข้อมูลเพื่อที่จะนำไปแสดงผลเป็นตัวอักษรวิ่งในรูปแบบต่างๆได้

2.1 ลักษณะโครงสร้างของแผงไฟอักษรวิ่ง

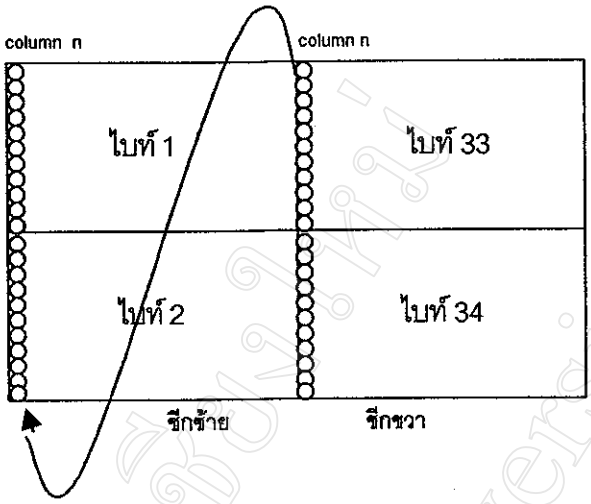
ระบบแผงไฟอักษรวิ่งมีโครงสร้างเป็น 4 บอร์ดย่อยที่ต่อเรียงอันดับแบบอนุกรมกันซึ่งแต่ละบอร์ดจะมีคุณสมบัติเหมือนกันและสับเปลี่ยนกันได้ มีการจัดเรียงของหลอด LED เป็นแบบ dot matrix มีขนาด 16 row X 32 column สำหรับการควบคุมการทำงานนั้นแต่ละบอร์ดจะมีขั้วต่อสายสัญญาณเข้าเพื่อรับข้อมูลและสัญญาณควบคุมจากอุปกรณ์ภายนอก และมีขั้วต่อสายสัญญาณออก เพื่อใช้ในการส่งผ่านข้อมูลและสัญญาณควบคุมไปยังบอร์ดถัดไป

การบรรจุข้อมูลเข้าแผงสามารถส่งจากบอร์ดแรกถึงบอร์ดสุดท้ายได้ด้วยวิธีการขับข้อมูลเข้าทีละ 1 บิท แบบอนุกรมโดยผ่านสายนำส่งข้อมูล 1 เส้นที่ต่อแบบอนุกรมกันทุกบอร์ด ส่วนสายสัญญาณควบคุมอื่นๆ ได้ต่อแบบขนานกันเพราะทุกบอร์ดต้องการสัญญาณควบคุมเดียวกัน



รูปที่ 2.1 แสดงการเชื่อมบอร์ด LED เข้าด้วยกันเพื่อเพิ่มความยาว

ในแต่ละบอร์ดย่อยได้จัดแบ่งโครงสร้างตามลักษณะการทำงานออกเป็น 2 ซีก คือ ซีกซ้าย 16 columnและซีกขวา 16 columnโดยที่column 0 และcolumn 16 ต่อแบบอนุกรมและทำงานในเวลาเดียวกัน สำหรับ columnที่เหลือทั้งหมดต่อกันแบบอนุกรมตามลำดับของซีกซ้ายและซีกขวาจนครบทุกcolumnซึ่งการทำงานของแต่ละcolumnให้ผลเท่าเทียมกับการสแกน ด้วยอัตราส่วนการทำงาน 1/16 ข้อมูลที่ใช้ในการแสดงผล 1 ครั้งมีจำนวน 4 ไบท์ ดังรูปแสดงที่ 2.2



รูป 2.2 แสดงการแบ่งcolumnในการทำงานของ 1 บอร์ดย่อย

LED บนแผงไฟจะทำงานเมื่อข้อมูลบิทนั้นๆเป็นลอจิก "0"

คือ ลอจิก "0" แสดงถึงข้อมูลพร้อมที่ทำให้ LED สว่าง ON

ลอจิก "1" แสดงถึงข้อมูลพร้อมที่ทำให้ LED สว่าง OFF

2.2 การสื่อสารข้อมูลและการควบคุม

ในแต่ละบอร์ดของแผงไฟจะมีสายสัญญาณสำหรับสื่อสารข้อมูลและควบคุมการทำงานรวมทั้งหมด 8 เส้น ในที่นี้มีเพียง 1 เส้นซึ่งเป็นสายสัญญาณข้อมูล ส่วนที่เหลือทั้งหมดเป็นสายสัญญาณควบคุมการแสดงผล (ดูตามรูปที่ 2.3 ประกอบ ซึ่งมีรายละเอียดดังนี้)

(1) ASI สายส่งข้อมูลแบบอนุกรม 1 เส้น

ข้อมูลเป็นลอจิก "0" หมายถึง LED ON

ข้อมูลเป็นลอจิก "1" หมายถึง LED OFF

(2) ACK เป็นสายส่งสัญญาณนาฬิกา 1 เส้น

การทำงานมีหน้าที่ในการเลื่อนข้อมูลเข้าแผงไฟ สำหรับสายสัญญาณดังกล่าวนี้จะใช้ควบคู่กับการส่งข้อมูล โดยจะสั่งข้อมูลให้เข้าไป 1 บิท แล้วตามด้วย นาฬิกา 1 ครั้งเพื่อเป็นส่วนที่ให้จังหวะการเลื่อนข้อมูลเข้าแผงไฟ

(3) INHEX (INHIBIT ON/OFF) จำนวน 1 เส้น ใช้สำหรับควบคุมให้ LED ON/OFF

ถ้าลอจิกเป็น "0" แสดงถึง LED ON

ถ้าลอจิกเป็น "1" แสดงถึง LED OFF

(4) column select ใช้สำหรับควบคุมสัญญาณการเลือก column จำนวนสี่เส้น KAI, KBI, KCI, KDI ในการทำงานของป้ายแสดงผลเราสามารถเลือกได้ว่าจะให้columnใดทำงานและสามารถเลือกได้ทั้งหมด 16 column

(5) ASTB จำนวน 1 เส้น เป็นสายสัญญาณควบคุมการ latch/transfer data ในการแสดงผลซึ่งมีความไวต่อระดับ latch ที่ลอจิก "0" และ Transfer ที่ลอจิก "1"

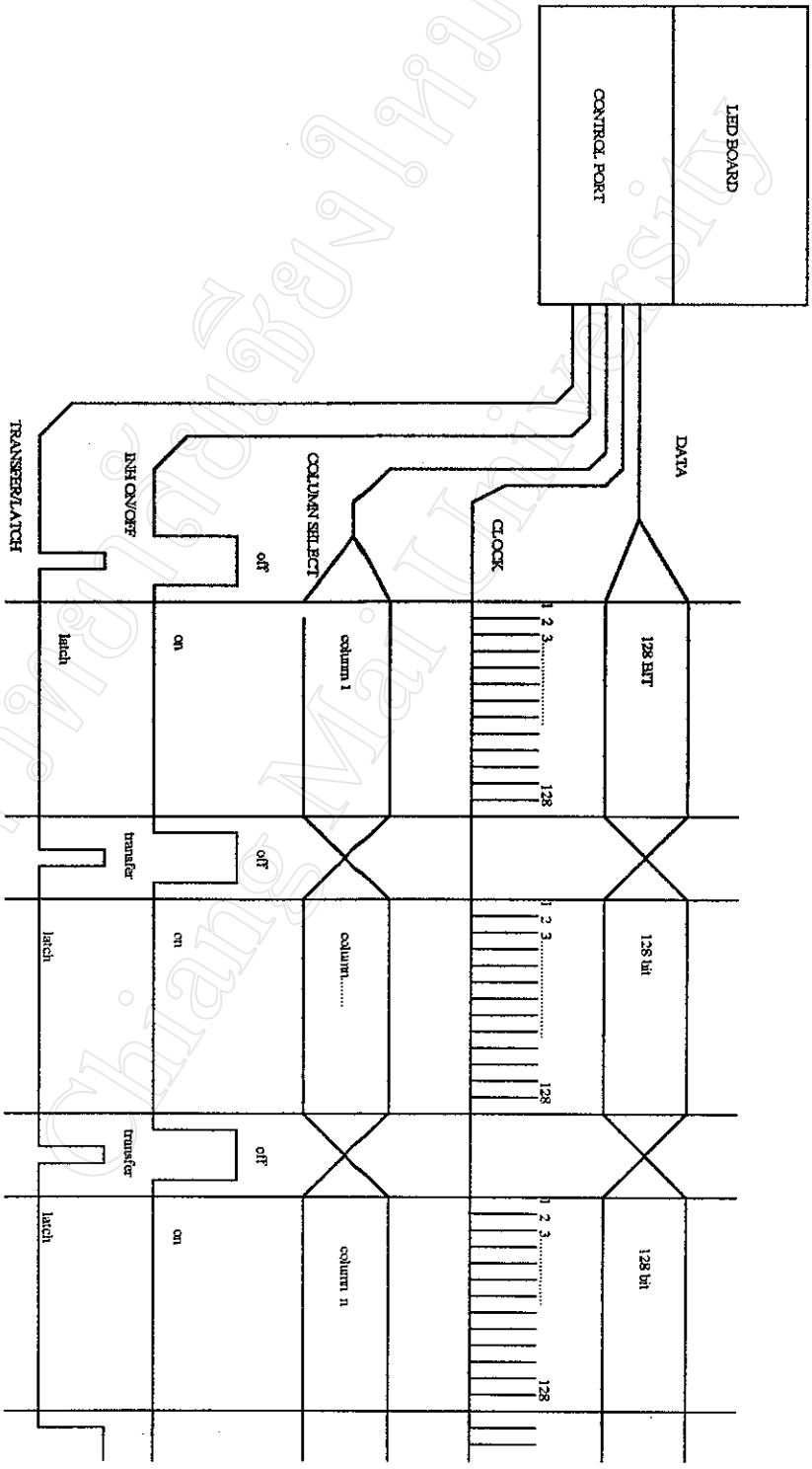
สำหรับสายควบคุมสัญญาณนี้ใช้เทียบกับสายสัญญาณ INHIBIT ON/OFF โดยที่ช่วงที่มีการ Transfer คือช่วงที่จะนำข้อมูลชุดใหม่แสดงผลแทนข้อมูลชุดเก่า จำเป็นต้องดับ (OFF) LED ของข้อมูลชุดเก่าชั่วขณะก่อนที่จะแสดงผลข้อมูลของชุดใหม่ ทั้งนี้เพื่อหลีกเลี่ยงการสว่างซ้ำซ้อนกันระหว่างข้อมูลชุดเก่าและใหม่ จากนั้นจึง transfer ข้อมูลชุดใหม่เพื่อแสดงผล

สำหรับเทคนิคปฏิบัติที่นำมาใช้ในช่วงเวลาของการเปลี่ยนแปลงข้อมูลจะเริ่มด้วยการสมมติว่า บนบอร์ดแสดงผลทุกบอร์ดพร้อมแล้ว จากนั้นลำดับของสัญญาณควบคุมจะเป็นดั่งนี้

LED ต้องดับก่อน INHEX (OFF) ลอจิก "1"

Transfer/ Latch ลอจิก "1" แล้วกลับเป็น "0"

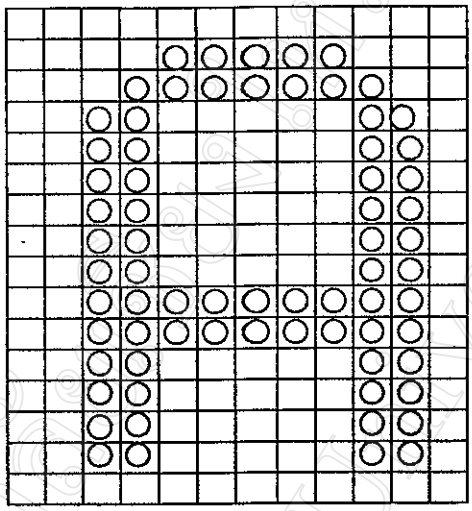
LED สว่าง INHEX (ON) ลอจิก "0"



รูปที่ 2.3 แสดงการทำงานของสัญญาณควบคุมต่าง ๆ

2.3 ขนาด, รูปแบบของตัวอักษรที่แสดงผล

แบบของตัวอักษรที่สร้างขึ้นเพื่องานวิจัยนี้มีขนาด 16 แถว x 9 columnหรือเทียบเท่ากับข้อมูล 18 ไบท์ ต่อ 1 ตัวอักษร ยกเว้น เลข 1 และอักษร I มี 10 ไบท์ และ 4 ไบท์ ตามลำดับ



รูป 2.4 แสดงรูปแบบ Dot matrix ของตัวอักษร

ตารางโครงสร้างของตัวอักษรทั้งหมดได้เก็บไว้ใน ความจำถาวรชนิดลบและเขียนใหม่ได้ (Erasable programable read only memory, EPROM) โดยเรียงตามลำดับของข้อมูลเมตริก ตัวอักษรทุกตัวตามลำดับของรหัส ASCII โดยเริ่มจาก 0 ที่มีรหัสเป็น 30 HEX

<u>รูปแบบการจัดเรียงข้อมูลตามรหัส ASCII ในตาราง</u>

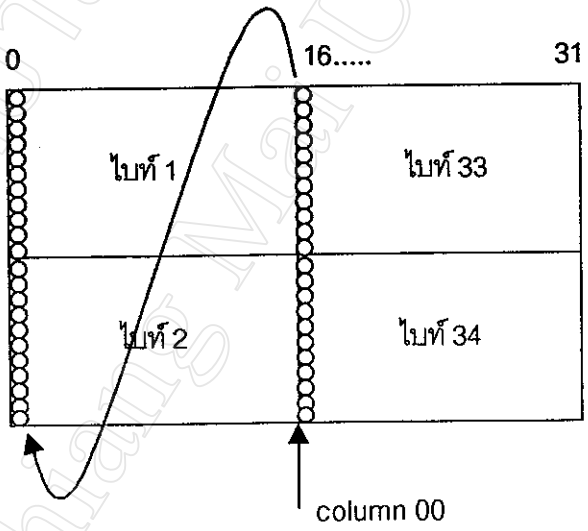
CHEXARACTER	ASC 11 CODE	ADDRESS	DATA
0	30 HEX	----	E0 07 C0 03 9F C1 9F 09 9C 39 90 F9 83 F9 C0 03 E0 07
1	31 HEX	----	F3 FF E7 FF CF FC 80 01 80 01
			-
			-
			-

2.4 การนำข้อมูลเข้าแผงไฟอักษรวิ่ง

เนื่องจากแผงไฟมีสายสัญญาณข้อมูลเพียง 1 เส้นเพื่อใช้ในการรับส่งข้อมูลแบบอนุกรม ข้อมูลจะเข้าไปยังแผงไฟทีละ 1 บิท โดยวิธีการผลักข้อมูล เมื่อเราส่งบิทแรกเข้าไปก่อนและส่งบิทที่ 2 ตามเข้าไป บิทที่ 2 จะผลักข้อมูลบิทแรกให้เลื่อนไปหนึ่งตำแหน่ง (1 บิท) และเมื่อเราให้ข้อมูลบิทใหม่เข้าไปอีก ข้อมูลบิทใหม่จะผลักข้อมูลบิทเก่าให้เลื่อนเดินหน้าไปอีกหนึ่งบิท จะเป็นลักษณะแบบนี้จนข้อมูลที่เรานำเข้าไปแสดงผลนั้นครบทุกบิท

ในหนึ่งบอร์ดย่อย เราสามารถเลือก column เพื่อที่จะส่งข้อมูลเข้าไปแสดงผลมีทั้งหมด 16 column เริ่มจาก column 0 ถึง column 15 ข้อมูลที่จะส่งเข้าไปแสดงผลมี 32 บิทต่อ 1 column หรือเท่ากับ 4 ไบท์ ทั้งหมด 16 column = 32 x 16 = 512 บิท หรือเท่ากับ 64 ไบท์

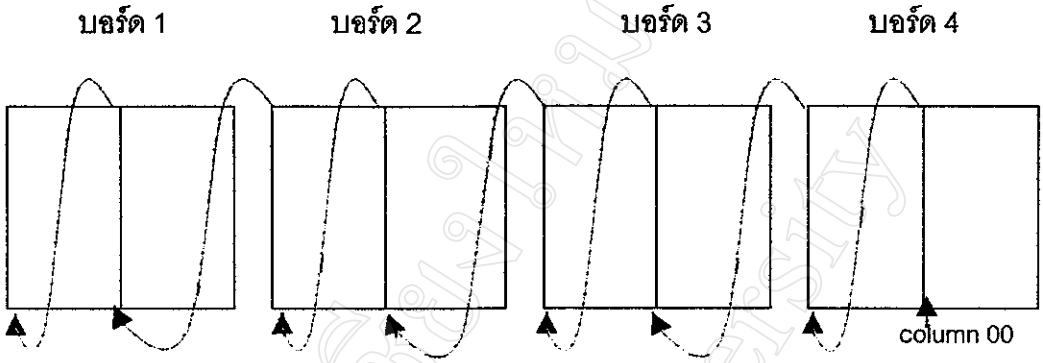
เลขที่ไบท์ที่แสดงในรูปที่ 5 เขียนตามผังของหน่วยความจำ ซึ่ง 1 บอร์ดใช้เนื้อที่ 64 ไบท์ โดยเริ่มนับจากไบท์ที่ 1 และ 2 ของ column 0 ถึงไบท์ 63 และ 64 ของ column 31



รูป 2.5 แสดงการนำข้อมูลเลื่อนเข้าcolumn 00 ซึ่งเป็นcolumnที่ 0 และcolumnที่ 16 ของหนึ่งบอร์ดย่อย

ในการทำงานจริงเรามี 4 บอร์ดย่อยที่ต่อแบบอนุกรมกัน การเลือก column เพื่อที่จะส่งข้อมูลเข้าไปแสดงผลนั้นสามารถเลือกได้ 16 column เหมือนกับการส่งข้อมูลในหนึ่งบอร์ด เนื่องจากสายควบคุมการเลือก column ได้ต่อขนานกันทุกบอร์ดเพื่อที่จะให้ทุกบอร์ดทำงานในเวลาเดียวกันได้ แต่ข้อมูลที่ส่งเข้าไปแสดงผลใน 1 column เท่ากับ 32 x 4 = 128 บิท หรือเท่ากับ 16

ไบท์ ทั้งหมด 16 column จำนวนข้อมูลที่จะส่งเข้าไปแสดงผลเท่ากับ 128 บิท x 16 = 2048 บิท หรือ 256 ไบท์



รูปที่ 2.6 แสดงการนำข้อมูลเข้าcolumnที่ 0 และcolumnที่ 16 ของ 4 บอร์ด

จากรูปที่ 6 ได้แสดงให้เห็นว่า column ที่ 0 และ column ที่ 16 ของแต่ละบอร์ดได้ต่อแบบอนุกรมกัน เราถือว่าเป็น column เดียวกัน และเรียก column นี้เป็นcolumn 00 สำหรับcolumnที่เหลือก็เป็นลักษณะเดียวกันทุกcolumnซึ่งมีทั้งหมด 16 column เริ่มจากcolumn 00 ถึง column 15

2.5 columnการแสดงผล

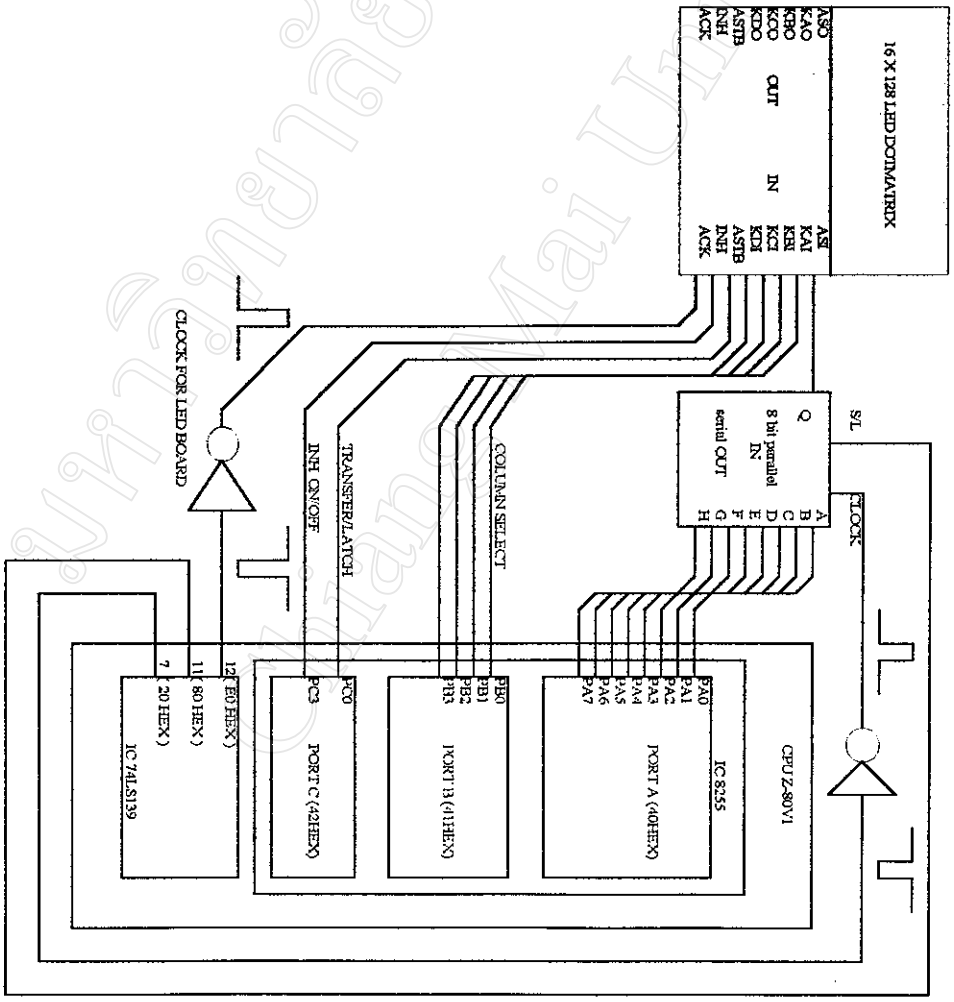
การแสดงผลของระบบแผงไฟอักษรวิ่ง มีcolumnการทำงานอยู่ที่การสแกน ซึ่งการสแกนนี้ทำได้ 2 วิธี คือการ สแกนแบบ แถว และการสแกนแบบ column แต่โครงสร้างของระบบแผงไฟที่ใช้ในการทำวิจัยนี้ได้ออกแบบไว้ให้ สแกน แบบ column

การสแกนแบบ column คือ จะส่งข้อมูลเข้าไปตาม column ที่เรากำหนดไว้ โดยวิธีการเลื่อนข้อมูลเข้าทีละ 1 บิทจนครบทุกบิท แล้วกำหนดให้ column นั้นทำงาน แต่ในความจริงแล้วในการที่จะแสดงผลเป็นรูปแบบของตัวอักษรได้นั้นจะต้องมีการสแกนทุก column ในระบบแผงไฟ ซึ่งมีรายละเอียดขั้นตอนการสแกนดังนี้คือ เริ่มจาก column 0 เราจะทำการส่งข้อมูลเข้าทีละบิทจนครบทุกบิท แล้วกำหนดให้ column 0 แสดงผล หลังจากนั้นทำการส่งข้อมูลเข้า column ถัดไปจนครบทุกบิท แล้วทำการแสดงผลแต่ที่สำคัญก่อนที่จะทำการแสดงผลใน column ถัดไปทุกครั้งต้องมีการเคลียร์ข้อมูลใน column แรกเสียก่อน ทำเช่นนี้เพื่อหลีกเลี่ยงการเสว่างซ้อนปรากฏเป็นเงาจางระหว่างสอง column ที่ติดกัน ทำเช่นนี้ไปจนกระทั่งครบทั้ง 16 column ก็จะเป็นการสแกนครบ 1 รอบ จากการทำงานของระบบแผงไฟจะเห็นได้ว่าเป็นการทำงานทีละ column ดังนั้นการสแกนในรอบหนึ่งๆต้องกำหนดให้ความถี่สูงพอที่จะทำให้การแสดงผลดูเหมือนว่าสว่างและ

ต่อเนื่องกัน โดยไม่มีการกระเพื่อมของหลอด LED ในเวลาแสดงผล โปรแกรมควบคุมการแสดงผลที่ได้เขียนขึ้นมา มีความถี่เท่ากับ 40 Hz พอดี

2.6 โครงสร้างวงจรควบคุมการทำงานทั้งระบบ

เนื่องจากแผลไฟอักษรวิ่ง ต้องการสัญญาณควบคุมการทำงานทั้งหมด 8 เส้น ในขณะที่พอร์ท I/O ของคอนโทรเลอร์บอร์ดที่นำมาใช้มีจำนวนจำกัด ทั้งนี้เพื่อให้อัตราการส่งข้อมูลเร็วพอที่จะไม่เห็นการกระพริบของตัวอักษรตอนแสดงผล ดังนั้นจึงได้ออกแบบวงจรเพิ่มเติมเพื่อการเชื่อมต่อระหว่างคอนโทรเลอร์บอร์ดกับระบบแผงไฟดังแสดงในผังวงจรรูปที่ 2.7



รูปที่ 2.7 แสดลผังการควบคุมการทำงานของระบบแผงไฟวิ่ง

(1) พอร์ท IC 8255

พอร์ท 40HEX (PA - PA_7) เป็นส่วนนำ data จาก controller board เข้าไปยัง IC 74LS165 ซึ่งเป็น shift register ที่ทำหน้าที่รับข้อมูลเข้า 8 บิท แบบขนาน และส่งออกแบบอนุกรมทีละ 1 บิท เข้าไปยังสายสัญญาณข้อมูลของแผงไฟซึ่งจะเป็นส่วนที่ทำให้เวลาในการส่งข้อมูลมีอัตราเร็วขึ้น ทำให้การทำงานมีความถี่สูงพอที่จะไม่เกิดการกระพริบของตัวอักษร

- พอร์ท 41HEX (PB_0 - PB_3) ใช้ต่อกับ สายสัญญาณควบคุมที่ใช้เลือก column KAI, KBI, KCI, KDI, ตามลำดับเพื่อเลือกหนึ่งใน 16 column โดยวิทีสะแกนขณะแสดงผล

- พอร์ท 42HEX : PC_0 ใช้ต่อเข้ากับสายสัญญาณควบคุม transfer / latch เพื่อที่จะทำการ transfer และ latch ข้อมูลในเวลาแสดงผล

PC_3 ใช้ต่อเข้ากับ Bus ควบคุม INHIBIT ON/OFFเพื่อที่ตะทำการ ON/OFF L8D ในเวลาแสดงผล

(2) สร้างพอร์ทสัญญาณเข้า-ออก เพิ่มเติม โดยใช้สัญญาณ $\overline{ps}$ จาก IC 74LS139

(E0HEX) ใช้ต่อกับสาย นาฬิกา เพื่อเป็นส่วนสร้าง นาฬิกา ให้กับการควบคุมการทำงานแผงไฟ

(20HEX) ใช้ต่อกับขา นาฬิกา ของIC 74LS165 เพื่อควบคุมการ shift data ออกทีละ 1 บิท แบบอนุกรมกับแผงไฟ

(80HEX) ใช้ต่อกับขา S/L ของ IC 74LS165 เพื่อควบคุมการ load ข้อมูลเข้าและ shift ข้อมูลออกตามจังหวะของ นาฬิกาที่ส่งให้

2.7 columnการทำงานของ Software

วิธีการแสดงผล ก่อนอื่นทำการจัดพื้นที่ไว้บนหน่วยความจำ RAM เพื่อพักข้อมูลที่ค้นมาจาก Table ไว้ชั่วคราวก่อนที่จะนำไปแสดงผล อย่างน้อยให้มีพื้นที่เท่ากับ 256 ไบท์ ซึ่งเท่ากับขนาดของบอร์ดสำหรับด้านบนและด้านล่างของ display window เราจัดให้มีพื้นที่ว่างไว้สำหรับการที่จะทำให้เป็นตัวอักษรวิ่งได้ การแสดงผลขั้นตอนที่ต้องทำก่อนคือ ต้องค้นหาข้อมูลที่จะนำไปแสดงผลจากรหัส ASCII เข้าไปเก็บไว้ใน display window ก่อนที่จะถึงขั้นตอนแสดงผล

แอดเดรส	การใช้ประโยชน์
8000	ที่ว่าง
8100	สำหรับรับข้อมูลที่ค้นมาจากตาราง
8200	ที่ว่าง
8300	set ค่าต่างที่สำคัญสำหรับการแสดงผล
8300 8302 8304 8306 8308	
9FFF	

รูปที่ 2.8 แสดงการใช้พื้นที่ของ RAM สำหรับทำหน้าต่างแสดงผล (display window)

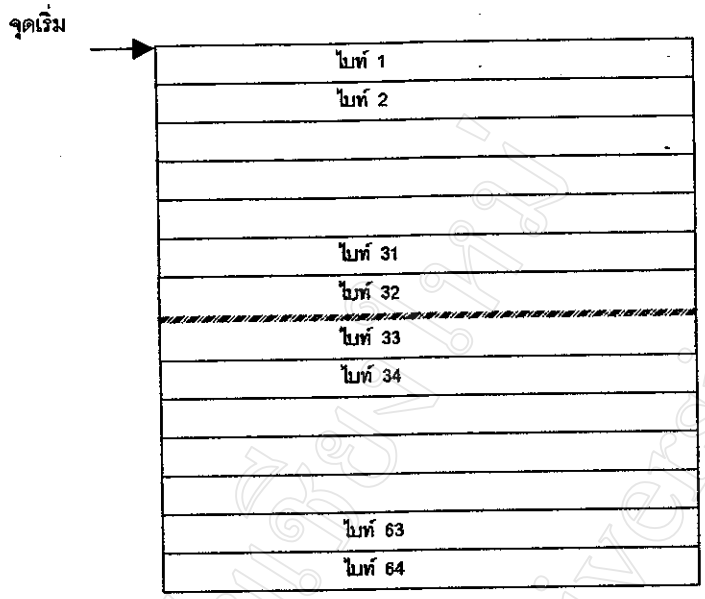
ส่วนที่ 1 เป็นพื้นที่สำหรับการแสดงผลที่ทำให้อักษรเลื่อนซ้าย,ขวา ด้วยการคัดลอกข้อมูลเข้ามาทีละชุดของตัวอักษร เก็บไว้ที่ แอดเดรส 8000-82FF

ส่วนที่ 2 ที่ แอดเดรส 8300-8308 ไว้สำหรับ set ค่าต่างๆที่จำเป็นในการแสดงผล

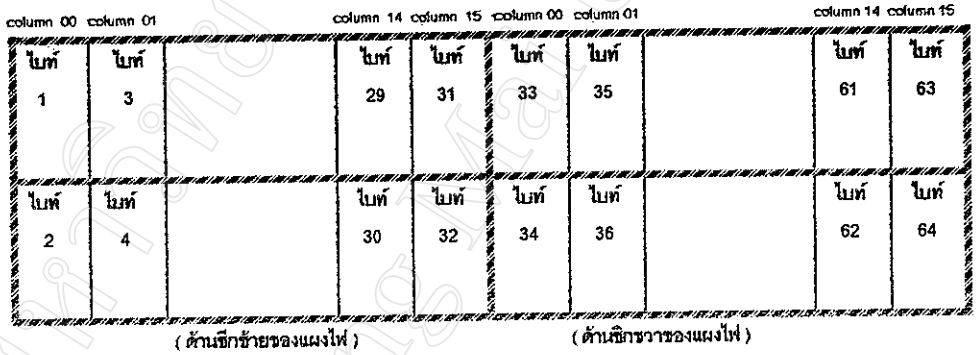
ส่วนที่ 2 ตั้งแต่ แอดเดรส 8400 ขึ้นไปไว้สำหรับเก็บข้อมูลที่ส่งเข้ามาจาก pc จำนวนข้อมูลที่ส่งเข้ามานั้นมีได้มากสุดไม่เกิน 256 byte ต่อหนึ่งชุดตัวอักษรตามขนาดความจุของแผงไฟ

การนำข้อมูลจาก display window เข้าบรรจุในแผงไฟ

เพื่อจะแสดงวิธีการจัดการให้เข้าใจโดยง่ายในตัวอย่างนี้จะเสนอวิธีการนำข้อมูลจาก splay window เข้าบรรจุในแผงไฟเพียง 1 บอร์ด ที่มีขนาด 16 row X 32 column ดังนั้นจำนวนข้อมูลใน display window ที่จะนำไปแสดงผลเท่ากับ 64 ไบท์ซึ่งมีขนาดความยาวเท่ากับ 1 บอร์ดพอดี



รูปที่ 2.9 แสดงการจัดเรียง ไบท์ข้อมูลใน display window ที่มีความยาวเท่ากับ 1 บอร์ด



รูปที่ 2.10 แสดงข้อมูลของ display window เมื่อจัดวางตามลำดับลงไปในแผงไฟอักษรวิ่ง

จากรูปที่ 2.10 เห็นได้ว่าหนึ่งบอร์ดย่อยจะแบ่งออกเป็น 2 ซีกเท่ากันมีขนาด 16 row X16 column และในแต่ละซีกจะมีการเชื่อมโยงระหว่าง column เข้าหากันแบบอนุกรม ซึ่งมีลำดับการเชื่อมโยง เช่น column ที่ 0 กับ columnที่ 16 , column ที่ 1 กับ column ที่ 17 ตามลำดับไปจนถึง column ที่ 15 กับ column ที่ 33 ดังนั้นในการเลือก column เพื่อที่จะนำข้อมูลเข้าไปแสดงผล เราสามารถเลือกได้ทั้งหมด 16 column จาก column 0 ถึง column ที่ 15, ข้อมูลจะเข้าตาม columnที่ถูกเลือกทางด้านซีกขวาก่อนที่จะทำการขับข้อมูลเข้าถึง column ทางด้านซีกซ้ายของบอร์ด รายละเอียดในการนำข้อมูลจาก display window เข้าไปยังแผงไฟ 1 บอร์ด มีดังนี้

ในขณะที่เราเลือกให้column ใดcolumnหนึ่งทำงานเช่น: เริ่มจากcolumn 0 ซึ่งเป็นcolumnที่ 0 และ columnที่ 16 ในแผงไฟให้อยู่ในภาวะทำงานแล้วเราจะทำการอ่านข้อมูลไบท์ที่ 1,2,33 และ 34 ออกจาก display window เข้าไปยัง columnดังกล่าว จนครบทุกบิทแล้วจึงให้ LED ON หลังจากนั้นจึงเปลี่ยนการทำงานไปที่column 01 คือ column ที่ 1 และcolumnที่ 17 แล้วทำการอ่านข้อมูลจากไบท์ที่ 3,4,35 และ 36 ออกไป แสดงผลทำเช่นนี้จนครบทุกcolumn เราจะเห็นได้ว่าในแต่ละบอร์ดจะเป็นแบบคอลัมน์คู่ ข้อมูลที่นำไปแสดงผลมีจำนวน 4 ไบท์ โดยมีการเรียงลำดับข้อมูลดังนี้

column 00 byte 1, byte 2, byte (7+32), byte (2+32)

column 01 : byte 3, byte 4, byte (3+32), byte (4+32)

column 02 : byte 5, byte 6, byte (5+32), byte (6+32)

,

,

column 15 : byte 31, byte 32, byte(31 + 32), byte(32 + 32)

จากการนำเสนอข้างต้นนั้นเป็นวิธีการนำข้อมูลเข้าเพียง 1 บอร์ด เท่านั้น ถ้าเป็น 4 บอร์ด ก็ทำในลักษณะเดียวกัน โดยเราจะนำข้อมูลเข้าไปแสดงผลทั้งหมด 16 ไบท์ต่อ 1 column ซึ่งจะเรียงลำดับการนำข้อมูลเข้าดังนี้,

column 00 : byte 1 , byte 2 , byte(1 + 32) , byte(2 + 32)
byte(1 + 64) , byte(2 + 64) , byte (1 + 96) , byte(2 + 96)
byte(1 + 128) , byte(2 + 128) , byte (1 + 160) , byte(2 + 160)
byte(1 + 192) , byte(2 + 192) , byte (1 + 224) , byte(2 + 224)

column 01 : byte 3 , byte 4 , byte(3 + 32) , byte(4 + 32)
byte(3 + 64) , byte(4 + 64) , byte (3 + 96) , byte(4 + 96)
byte(3 + 128) , byte(4 + 128) , byte (3 + 160) , byte(4 + 160)
byte(3 + 192) , byte(4 + 192) , byte (3 + 224) , byte(4 + 224)
| | | |

column 15 : byte 31 , byte 32 , byte(31 + 32) , byte(32 + 32)
byte(31 + 64) , byte(32 + 64) , byte (31 + 96) , byte(32 + 96)
byte(31 + 128) , byte(32 + 128) , byte (31 + 160) , byte(32 + 160)
byte(1 + 192) , byte(2 + 192) , byte (1 + 224) , byte(2 + 224)

เมื่อเรามีข้อมูลที่จะนำไปแสดงผลเก็บไว้ใน display window แล้ว ขั้นตอนต่อไปจะเป็นการนำข้อมูลออกไปแสดงผล ในการแสดงผลขึ้นอยู่กับขั้นตอนของการสแกน

<u>ลำดับขั้นตอนในการสแกน 1 รอบ (ดูรูป 2.7 ประกอบ)</u>

1.) นำข้อมูลเข้าไปยังแผงไฟอักษรวิ่ง

การนำข้อมูลเข้าไปยังแผงไฟ เราจะให้สัญญาณที่พอร์ทเลขที่ 80 HEX เป็น "0" เพื่อเป็นการ นำข้อมูลขนาด 8 บิทขา shift/load ของ เข้าไปยัง IC 74LS165 ผ่านทาง พอร์ท A (PA0 →PA7) หลังจากนั้น ให้สัญญาณที่ พอร์ท 80HEX กลับขึ้นเป็นลอกจิก"1" เพื่อเป็นการนำข้อมูลออกจาก Shift Register เข้าไปยังแผงไปทีละ 1 บิทโดยผ่าน bus data ตามจังหวะของสัญญาณนาฬิกา และ shift clock (EOHEX และ 20HEX)

2.) เลือก column ที่จะทำการส่งข้อมูลเข้าแสดงผล

สายควบคุมการทำงานของcolumnมีทั้งหมด 4 เส้น และสามารถที่จะเลือก column ให้ทำงานได้ทั้งหมด 16 column เริ่มจาก column 0 ถึงcolumn 15

3.) การแสดงผลทีละ column

การแสดงผลแต่ละ column มีลำดับขั้นตอนของสัญญาณควบคุมดูตามรูป 11 ซึ่งมีรายละเอียดดังนี้ สิ่งแรกที่ต้องทำคือ เราต้องสั่งให้ LED ดับก่อน โดยส่งสัญญาณ 08HEX ไปยัง พอร์ท C_3 หลังทำการเลือก column แรกทำงานจากนั้นใช้พอร์ท C_0สำหรับผลิตสัญญาณ transfer และ off โดยการส่งสัญญาณ 09HEX เพื่อเป็นการ ดับ LED และ transfer ข้อมูลเข้าไปในเวลาเดียวกัน หลังจากนั้นเป็นขั้นตอนการบรรจุข้อมูลเข้าไปใน column ที่ถูกเลือกไว้ โดยผ่านสายส่งสัญญาณข้อมูลอนุกรมเข้าไปทีละ 1 บิท ตามด้วยสัญญาณ นาฬิกา 1 ลอจิก จนครบ 128 บิท หลังส่งลอจิก 00 HEX ไปยัง พอร์ท C_3อีกครั้ง เพื่อ latch ข้อมูลที่ค้างไว้และทำการเปิดการแสดงผล(LED ON) ในเวลาเดียวกัน สำหรับ column ถัดไปเราสามารปฏิบัติได้เช่นเดียวกันกับ column แรกทุกประการ เราจะทำเช่นนี้จนครบทุก column ก็จะเป็นการสแกนครบ 1 รอบ

<u>ข้อสังเกต</u>: การที่จะแสดงผลในcolumnถัดไปนั้นทุกครั้ง ต้องสั่งให้ LED ในcolumnที่กำลังแสดงผลอยู่นั้น อยู่ในภาวะ off ก่อน เพื่อไม่ให้เป็นการรบกวนข้อมูลที่จะแสดงผลในcolumnถัดไป จากรูปที่ 11 เราสามารถที่จะคำนวณความถี่ สูงสุดของการทำงานของระบบได้ ดังนี้:

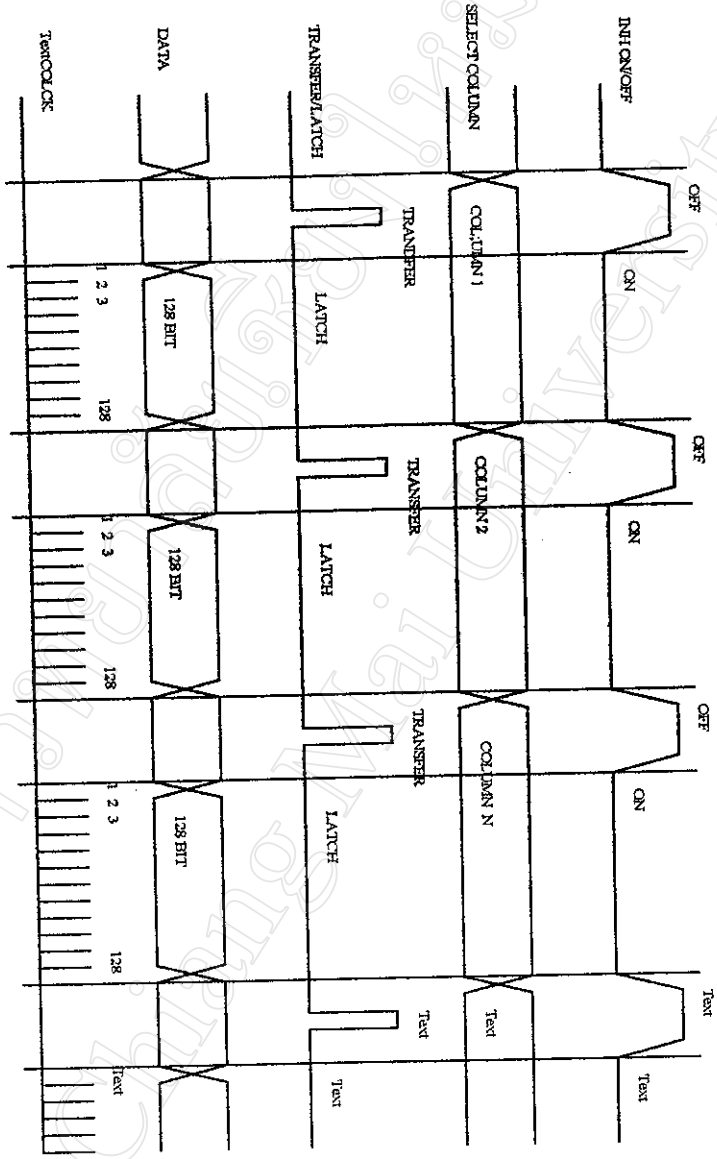
ช่วงเวลาของการทำงาน 1 column = เวลาในการส่งข้อมูล 128 บิท +เวลาการปิด LED เพื่อรับข้อมูล และปิด LED เพื่อแสดงผล

= 0.256 + 1.1 + 0.006 = 1.362 ms.

เวลาของการสแกน 1 รอบ = 16 column x1.362 ms. = 21.792 ms.

การทำงาน 1 รอบใช้เวลา 21.792 ms. นอกจากเวลาการนำข้อมูลเข้าสู่บอร์ดแล้ว ยังต้องใช้เวลาอีกส่วนหนึ่งเพื่อเคลื่อนย้ายข้อมูลในหน่วยความจำ ซึ่งเป็นการจัดเตรียมข้อมูลก่อนทำการสแกน การทำงานส่วนนี้ใช้เวลาประมาณ 4 ms ต่อรอบดังนั้นความถี่สูงสุดของการแสดงผลจะประมาณ 1/(21+4) ms หรือประมาณเท่ากับ 40 Hz ซึ่งเป็นความถี่พอที่สายตาของมนุษย์ไม่สามารถมองเห็นการกระพริบของตัวอักษรได้

มหาวิทยาลัยเชียงใหม่ Chiang Mai University



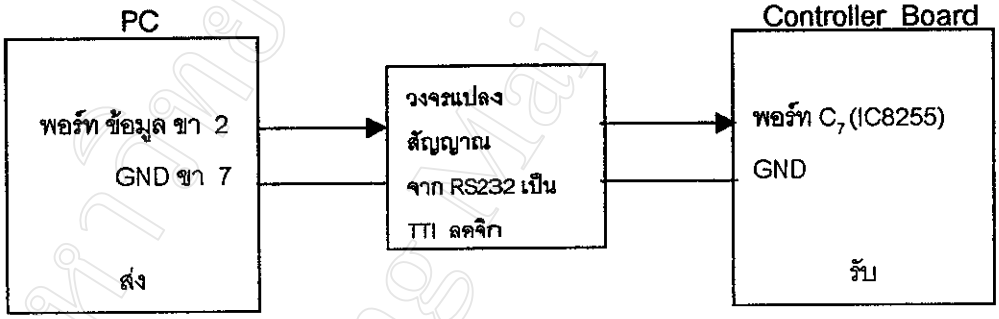
รูปที่ 2.11 แสดง time diagram ของการสแกน

2.8 การทำให้เป็นตัวอักษรวิ่ง

จากหัวข้อ 2.7 เราได้แบ่ง display window ไว้เป็น 3 ส่วน, ส่วนที่ 1 เป็นพื้นที่ไว้สำหรับรับข้อมูลมาแสดงผลแบบวิ่ง โดยจะคัดลอกข้อมูล จากส่วนที่ 2 เข้ามาทีละชุดของตัวอักษรแล้วทำการอ่านข้อมูลเพื่อจะนำไปแสดงผล โดยเริ่มอ่านจากจุด จุดเริ่ม ที่เรากำหนดขึ้นไปเรื่อยๆ จนถึง ไบท์สุดท้าย แล้วจะมองเห็นเป็นตัวอักษรแสดงบนบอร์ด การที่จะทำให้เป็นตัวอักษรวิ่งได้นั้นใช้วิธีการเลื่อนตำแหน่งข้อมูล การกำหนดจุด จุดเริ่ม ใหม่เพื่อที่จะทำการอ่านข้อมูลไปแสดงผลในรอบต่อไป ดังนั้นในการสแกน 1 รอบ เราจะกำหนดจุด จุดเริ่ม ไว้ เมื่อจะให้เป็นตัวอักษรวิ่งทำได้โดยการเปลี่ยน จุดเริ่ม ใหม่โดยเพิ่มหรือลดตำแหน่งไปอีก 2 byte แล้วจึงทำการ สแกน ใหม่ สำหรับในกรณีที่จะให้ตัวอักษรวิ่งเร็วหรือช้าขึ้นอยู่กับการกำหนดจำนวนรอบของการสแกนสำหรับจุดเริ่มจุดเดิม หรือจุดเดียวกัน

2.9 การส่งข้อมูลจาก PC ไปยัง Controller board

การส่งข้อมูลเพื่อที่จะเปลี่ยนชุดข้อความจากการแสดงผล ได้ใช้ PC เป็นตัวส่งข้อมูล ซึ่งมีโครงสร้างการเชื่อมต่อของระบบอุปกรณ์ดังนี้



รูปที่ 2.12 แสดงวงจรการเชื่อมต่อระหว่างตัวรับ Controller กับ PC

การส่งข้อมูลจาก PC ได้ส่งผ่านพอร์ท com (RS232) เนื่องระบบสัญญาณไฟที่พอร์ทสัญญาณเข้า-ออก ของตัวส่ง PC และตัวรับ Controller มีความแตกต่างกันเช่น พอร์ท Com (RS232) สัญญาณที่ส่งออกมาเป็น

ลอจิก "0" จะมีสัญญาณไฟ -9 Volt

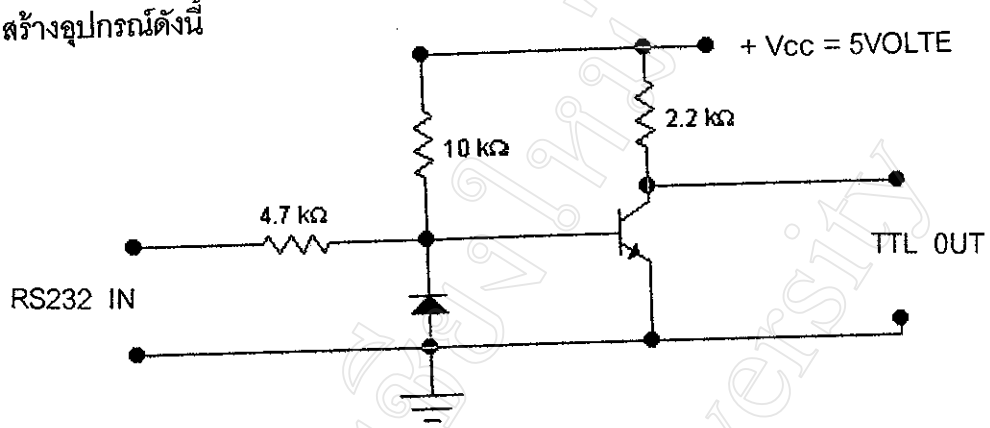
ลอจิก "1" จะมีสัญญาณไฟ +9 Volt

บน Controller ใช้ พอร์ทสัญญาณเข้า-ออก ของ IC8255 ทำงานเป็น TTL ลอจิก

ลอจิก "0" จะมีสัญญาณไฟ 0 Volt

ลอจิก "1" จะมีสัญญาณไฟ 5 Volt

ดังนั้นเพื่อให้สอดคล้องกับการทำงานที่เป็นระบบของอุปกรณ์ตัวรับและตัวส่งจึงได้สร้างวงจรแปลงสัญญาณขึ้นมาเพื่อทำทำหน้าที่แปลงสัญญาณจาก RS232 ลอจิก มาเป็นแบบ TTL ลอจิก ซึ่งมีโครงสร้างอุปกรณ์ดังนี้



รูปที่ 2.12 แสดงวงจรแปลงสัญญาณจาก RS232 ลอจิก เป็น TTL ลอจิก

วิธีการส่งข้อมูลของกรณีนี้ ในขั้นแรกได้โปรแกรมไว้บน PC เป็นสองส่วนคือส่วนของโปรแกรมแรกส่งข้อมูลจาก PC ออกไปเป็นแบบนุกรม ทีละ 1 บิท ตามรหัสเลขฐาน 2 ของตัวอักษรหรือรูปภาพที่สร้างขึ้นซึ่งการทำงานของโปรแกรมนี้จะทำการส่งข้อมูลที่ start bit เป็นลอจิก "0" หลังจากสิ้นสุดการส่งข้อแล้วจะกลับไปเป็น ลอจิก "1" อีกครั้ง

ส่วนโปรแกรมที่ 2 เป็นโปรแกรมสร้างบอร์ดแสดงผลจำลองให้แสดงผลบนจอ PC ซึ่งมีขนาด 16 แถว x 128 column เหมือนกับบอร์ดแสดงผลจริง ข้อความที่จะนำมาแสดงผลสามารถสร้างลงไปในบอร์ดจำลองก่อน เมื่อได้รูปแบบของข้อมูลตามที่ต้องการ ต่อไปเป็นขั้นตอนการนำไปแสดงผล ข้อมูลที่ไปปรากฏบนบอร์ดแสดงผลจะแสดงตามตำแหน่งและตามรูปแบบที่เราสร้างขึ้นบนบอร์ดจำลองทุกประการ (โปรแกรมที่กล่าวมานี้ใช้ภาษา Quick basic)

ส่วนของตัวรับ Controller ได้เขียนโปรแกรมเป็นภาษาเครื่องแอสเซมบลี เก็บไว้ใน EPROM ซึ่งเป็นโปรแกรม ตรวจสอบข้อมูล, start bit และ stop bit โดยวิธีเฝ้าดู (watch dog) เพื่อคอยตรวจสอบดูว่ามีข้อมูลส่งเข้ามาหรือไม่ ถ้ามีข้อมูลส่งเข้ามาการทำงานของโปรแกรมcolumnจะกระโดดออกไปทำงานที่โปรแกรมรับข้อมูลแล้วนำไปเก็บไว้ใน display windows ส่วนที่ 2 ของ RAM ตามแอดเดรส ที่กำหนดไว้ จากนั้นก็เป็นขั้นตอนการนำข้อมูลไปแสดงผล