

คู่มือการใช้งานบอร์ด CP-JR LPT v1.0 และ ET-BASE LPT v1.0



เขียนโปรแกรมควบคุมไอซี 8255 ผ่านเครื่องคอมพิวเตอร์ PC
ด้วย Visual Basic 6.0

ภายใน CD-Rom บรรจุไฟล์ตัวอย่างโปรแกรม VB6.0 พร้อม
คำอธิบาย

คำนำ

Preface

พอร์ตเครื่องพิมพ์ (Parallel Port) หรือ หลายคนอาจรู้จักในชื่อของพอร์ตปρί้นท์เตอร์ ซึ่งปัจจุบัน พอร์ตนี้ไม่ได้ใช้สำหรับต่อเครื่องพิมพ์เพียงอย่างเดียว แต่มันสามารถนำมาประยุกต์ใช้สำหรับการควบคุมอุปกรณ์ไฟฟ้า หรือ สิ่งต่างๆ ได้อีกด้วย โดยการเขียนโปรแกรมเพื่อส่งงานขาต่างๆ ของพอร์ตเครื่องพิมพ์ ซึ่งในเนื้อหาของคู่มือนี้จะอ้างอิงการเขียนโปรแกรมด้วย Visual Basic 6.0 เท่านั้น

เนื้อหาในหนังสือคู่มือนี้จะอ้างอิงกับบอร์ด 2 รุ่น คือ CP-JR LPT V1.0 และ ET-BASE LPT V1.0 ซึ่งบอร์ดทั้ง 2 รุ่นนี้จะมีวงจร และ การทำงานที่เหมือนกัน คือ เป็นบอร์ดควบคุมไอซี 8255 ด้วยพอร์ตเครื่องพิมพ์ ,เพียงแต่บอร์ดรุ่น ET-BASE LPT จะมีจัมเปอร์เลือกแหล่งจ่ายไฟ 5 VDC จากภายใน หรือ ภายนอกได้ด้วย

สุดท้ายนี้ผู้เขียน และ ทีมงาน ETT หวังเป็นอย่างยิ่งว่าหนังสือคู่มือ และ บอร์ดทดลอง CP-JR LPT/ET-BASE LPT V1.0 นี้จะช่วยให้ผู้อ่าน และ ผู้สนใจเข้าใจการทำงานของพอร์ตขนาน (Parallel Port) และ การต่อร่วมกับไอซี 8255 มากยิ่งขึ้น และ หากผู้อ่านมีข้อคิดเห็น หรือ มีข้อท้วงติงเกี่ยวกับเนื้อหาภายในหนังสือคู่มือฉบับนี้ สามารถให้ข้อคิดเห็นผ่านทาง e-mail : sales@etteam.com ซึ่งทางผู้เขียน และ ทีมงาน ETT จะขอโน้มรับไว้เพื่อการปรับปรุงในการต่อไป

จักรพันธ์ จิตรทรัพย์

14 กุมภาพันธ์ 2548

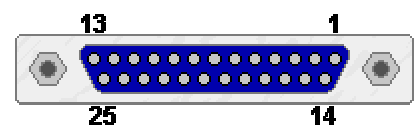
สารบัญ

Contents

บทที่ 1 ทฤษฎีการบนบอร์ด CP-JR LPT V1.0 /ET-BASE LPT V1.0	หน้า 1
บทที่ 2 ทฤษฎีพอร์ตขนาน	หน้า 6
บทที่ 3 ทฤษฎีไอซี 8255	หน้า 14
บทที่ 4 Inpout32.DLL ไลบรารีสำหรับติดต่อพอร์ตขนาน	หน้า 18
ภาคผนวก ก. การเซตอัป BIOS สำหรับเลือกพอร์ต LPT	
ภาคผนวก ข. การเซตอัปค่าต่างๆ ก่อนการใช้งาน VB6.0	
ภาคผนวก ค. วงจร Schematic บอร์ด CP-JR LPT/ET-BASE LPT V1.0	

1

ทรัพยากรบนบอร์ด

CP-JR LPT V1.0 / ET-BASE LPT V1.0

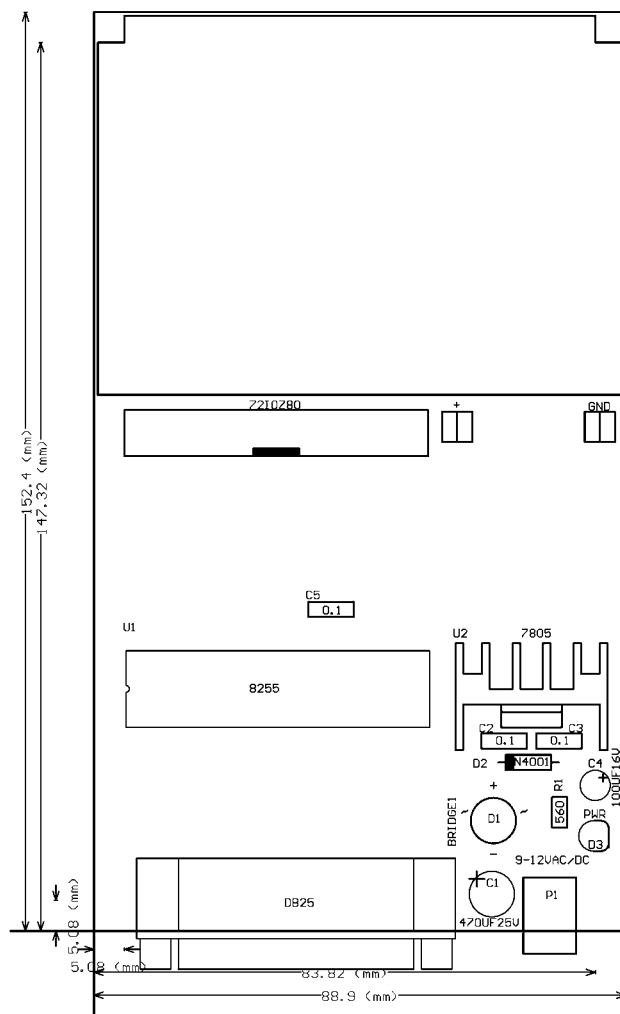
บอร์ด CP-JR LPT v1.0 และ ET-BASE LPT v1.0 เป็นบอร์ดขยายพอร์ต I/O ด้วยไอซี 8255 ซึ่งควบคุมการทำงานผ่านทางพอร์ตเครื่องพิมพ์ (Parallel Port) โดยในส่วนของการจัดวงจรทาง Hardware ของบอร์ดทั้ง 2 รุ่นนี้จะอ้างอิงกับโหมด EPP (Enhanced Parallel Port) ของพอร์ตเครื่องพิมพ์

บอร์ด CP-JR LPT v1.0 และ ET-BASE LPT v1.0 ทั้ง 2 รุ่นนี้จะนำท่านผู้อ่านสู่การควบคุมอุปกรณ์อิเล็กทรอนิกส์ หรือ อุปกรณ์ไฟฟ้า ผ่านทางพอร์ตเครื่องพิมพ์ (Printer port) ของคอมพิวเตอร์ ซึ่งจากเดิมหลายคนเคยผ่านการควบคุมด้วยไมโครคอนโทรลเลอร์มาบ้างแล้ว เช่น MCS-51, PIC หรือ Z-80. การควบคุม หรือ ส่งงาน I/O Port ผ่านทางคอมพิวเตอร์จะมีข้อดีตรงที่หน้าต่างโปรแกรมมีความสวยงาม ผู้ใช้สามารถมองเห็น และ เข้าใจการทำงานของโปรแกรมได้โดยง่าย

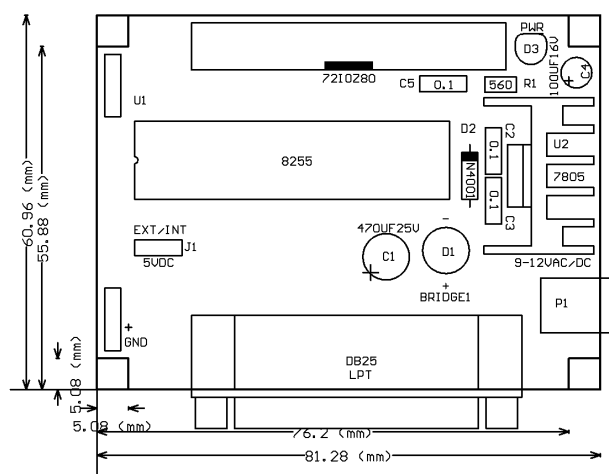
บอร์ด 2 รุ่นนี้จะมี วงจร และ การทำงานหลักๆ ที่เหมือนกัน แต่บอร์ดรุ่น ET-BASE LPT V1.0 จะมีจัมเปอร์สำหรับเลือกแหล่งจ่ายไฟ 5 VDC จากภายนอกได้ด้วย เนื่องจากว่าบอร์ดรุ่น ET-BASE LPT V1.0 ถูกออกแบบมาให้สามารถต่อใช้งานร่วมกับชุดฝึก ET-BASIC I/O V1.0 ได้ด้วย ดังนั้น ผู้อ่านสามารถดึงแรงดันไฟ 5 VDC จากบอร์ด ET-BASIC I/O V1.0 มาใช้ได้

การพัฒนาโปรแกรมในหนังสือคู่มือนี้จะอ้างอิงกับการเขียนโปรแกรมด้วย Microsoft Visual Basic 6.0 ซึ่งผู้อ่านจะต้องทำการติดตั้งโปรแกรมไว้บนเครื่อง PC เป็นที่เรียบร้อยแล้ว





รูปแสดงลักษณะโครงสร้างบอร์ด CP-JR LPT v1.0



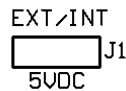
รูปแสดงลักษณะโครงสร้างบอร์ด ET-BASE LPT v1.0

ทรัพยากรบนบอร์ด CP-JR LPT v1.0/ ET-BASE LPT v1.0

ทรัพยากรบนบอร์ด CP-JR LPT v1.0 นั้นหลักๆ ได้แก่ ขั้วพอร์ต DB25 ซึ่งใช้สำหรับต่อระหว่างเครื่องคอมพิวเตอร์ กับ บอร์ด CP-JR LPT v1.0 , ขั้ว 34 Pin 72IOZ80 ใช้เป็นเอาต์พุตสำหรับต่อกับโหนดซึ่งพอร์ต PA, PB, PC ของไอซี 8255 จะถูกต่อออกมาที่ขั้ว 34 Pin นี้ , จัมเปอร์ EXT/INT จะมีเฉพาะบอร์ดรุ่น ET-BASE LPT V1.0 เท่านั้น ซึ่งใช้เลือกแหล่งจ่ายไฟ 5 VDC

จัมเปอร์ EXT/INT (สำหรับบอร์ด ET-BASE LPT V1.0 เท่านั้น)

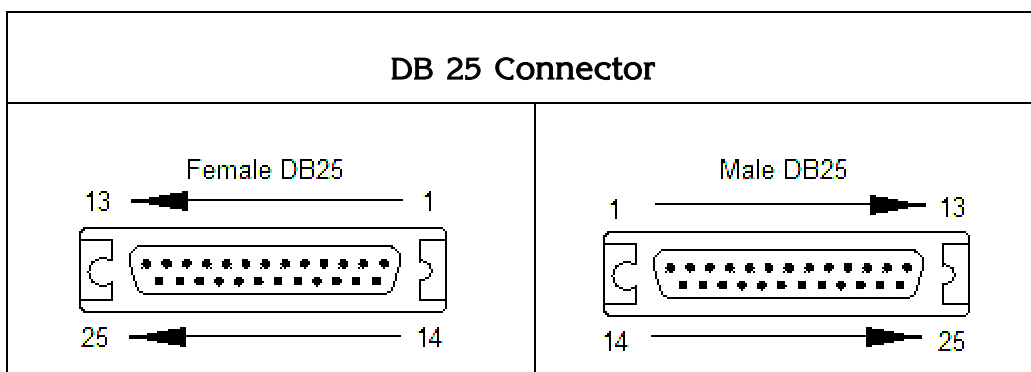
จัมเปอร์ EXT/INT ใช้สำหรับเลือกแหล่งจ่ายไฟ 5VDC จากภายนอก หรือ แหล่งจ่ายไฟจากภายใน ซึ่งจะรับไฟ 9 – 12 VAC/DC แล้วแปลงเป็น 5VDC



รูปแสดงจัมเปอร์ EXT/INT

พอร์ตขนาน (Parallel Port)

พอร์ตขนาน หรือ พอร์ตเครื่องพิมพ์มีขาทั้งหมด 25 ขา (D- type) ซึ่งผู้อ่านจะต้องต่อสายจากเครื่องคอมพิวเตอร์ PC มาที่ขั้วต่อนี้ ซึ่งการดูตำแหน่งขาแสดงดังภาพด้านล่าง



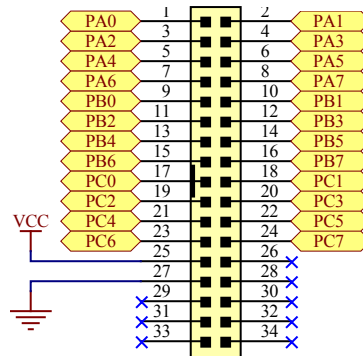
รูปแสดงการดูขาของพอร์ตขนาน (Parallel Port)

ตารางแสดงชื่อขาของพอร์ตขนาน (Parallel Port) ในโหมดการทำงานต่างๆ

Standard Parallel Port (SPP)			Enhanced Parallel Port (EPP)		Extended Capabilities Port (ECP)	
Signal Name	Centronics (36-pin)	DB-25 (25-pin)	Signal Name	DB-25 (25-pin)	Signal Name	DB-25 (25-pin)
Strobe	1	1	Write	1	Host CLK	1
Data 0	2	2	Data 0	2	Data 0	2
Data 1	3	3	Data 1	3	Data 1	3
Data 2	4	4	Data 2	4	Data 2	4
Data 3	5	5	Data 3	5	Data 3	5
Data 4	6	6	Data 4	6	Data 4	6
Data 5	7	7	Data 5	7	Data 5	7
Data 6	8	8	Data 6	8	Data 6	8
Data 7	9	9	Data 7	9	Data 7	9
Acknowledge	10	10	Interrupt	10	Periph CLK	10
Busy	11	11	Wait	11	Periph ACK	11
Paper End	12	12	Spare	12	Ack Reverse	12
Select	13	13	Spare	13	X-Flag	13
Auto Line Feed	14	14	Data Strobe	14	Host ACK	14
Error	32	15	Spare	15	Periph Request	15
Initialize Printer	31	16	Reset	16	Reverse Request	16
Select Input	36	17	Address Strobe	17	1284 Active	17
Strobe Return	19	18	Ground	18	Ground	18
Data 0 Return	20	19	Ground	19	Ground	19
Data 1 Return	21	19	Ground	20	Ground	20
Data 2 Return	22	20	Ground	21	Ground	21
Data 3 Return	23	20	Ground	22	Ground	22
Data 4 Return	24	21	Ground	23	Ground	23
Data 5 Return	25	21	Ground	24	Ground	24
Data 6 Return	26	22	Ground	25	Ground	25
Data 7 Return	27	22				
Acknowledge Return	28	24				
Busy Return	29	23				
Paper End Return	28	24				
Select Return	28	24				
Auto Line Feed Return	30	25				
Error Return	29	23				
Initialize Printer Return	30	25				
Select Input Return	30	25				
Shield	33	N/C				
N/C	34	N/C				
N/C	35	N/C				

พอร์ต 72IOZ80

พอร์ต 72IOZ80 เป็นพอร์ตขนาด 34 Pin ซึ่งแต่ละขาจะเป็นขาเอาต์พุตของ IC 8255 ซึ่งได้แก่ PA, PB, PC รวมถึงแหล่งจ่ายไฟ +5 V และ GND ด้วย ดังแสดงในภาพด้านล่าง

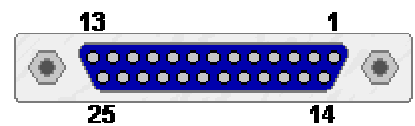


รูปแสดงชื่อขาต่างๆ บนพอร์ต 72IOZ80



2

ทฤษฎีพอร์ตขนาน



สิ่งแรกก่อนที่จะศึกษาพอร์ตขนาน (Parallel Port) คือ การเรียนรู้วิธีที่จะใช้ประโยชน์สูงสุดจากมันได้อย่างไร, รวมถึงวิธีการติดตั้งพอร์ต และ วิธีการเลือกโหมดการใช้งานต่างๆ ซึ่งเนื้อหาในบทที่ 2 นี้จะกล่าวถึงสิ่งจำเป็นต่างๆ สำหรับการใช้งานพอร์ตขนาน (Parallel Port)

คำจำกัดความของพอร์ต

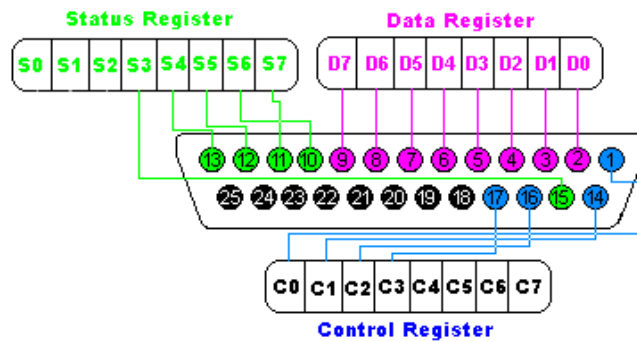
พอร์ตขนาน (Parallel Port) คือ อะไร ? ในโลกของคอมพิวเตอร์, พอร์ตๆ หนึ่งจะประกอบไปด้วยกลุ่มของสายสัญญาณหลายๆ เส้นมารวมกัน ซึ่งพอร์ตต่างๆ เหล่านี้จะใช้ในการแลกเปลี่ยนข้อมูลระหว่างพอร์ต (PC) กับ อุปกรณ์ต่างๆ ที่ต่ออยู่ด้วยกัน

พอร์ตขนาน (Parallel Port) คือ พอร์ตที่มีการส่งถ่ายข้อมูลหลายๆ บิตในเวลาเดียวกัน ในขณะที่พอร์ตอนุกรม (Serial Port) จะมีการส่งถ่ายข้อมูลทีละ 1 บิต ซึ่งจะส่งผลให้การส่งถ่ายข้อมูลของพอร์ตขนานนั้นมีความรวดเร็วกว่าพอร์ตอนุกรมแต่การใช้งานพอร์ตขนานจะต้องใช้จำนวนสายสัญญาณมากกว่า

เนื้อหาภายในบทนี้จะสามารถใช้งานได้กับคอมพิวเตอร์ PC หรือ คอมพิวเตอร์ส่วนบุคคลโดยทั่วไป ซึ่งอ้างอิงกับมาตรฐานของ IBM PC เนื่องจากคอมพิวเตอร์ส่วนบุคคลส่วนใหญ่จะพัฒนามาจากมาตรฐาน IBM อยู่แล้ว

พอร์ตขนาน (Parallel Port) ของคอมพิวเตอร์ในยุคแรกๆ จะมี 8 บิตที่ทำหน้าที่เป็นเอาต์พุต, มี 5 บิตทำหน้าที่เป็นอินพุต และ มี 4 บิตที่สามารถเป็นได้ทั้งอินพุต/เอาต์พุต (Bidirectional lines) แต่ปัจจุบันคอมพิวเตอร์ส่วนบุคคล (PCs) นั้นสามารถใช้งานเอาต์พุต 8 บิตให้เป็นอินพุต 8บิต ได้อีกด้วย ซึ่งทำให้อุปกรณ์ที่ต่อรอบนอกคอมพิวเตอร์ (Peripherals Devices) เช่น Scanners, drives และ อุปกรณ์อื่นๆ สามารถส่งข้อมูลกลับมาให้กับคอมพิวเตอร์ PC ได้ด้วย และ ทำให้การส่งถ่ายข้อมูลนั้นมีความรวดเร็วขึ้น

ก่อนหน้านี้, พอร์ตขนาน (Parallel Port) ถูกออกแบบให้เป็น Printer Port หรือ พอร์ตเครื่องพิมพ์เพียงอย่างเดียว ซึ่งผู้อ่านจะเห็นได้จากชื่อของสายสัญญาณต่างๆ ที่อยู่บนพอร์ตซึ่งจะสื่อถึงการใช้งานได้อย่างทันที เช่น PaperEnd, AutoLineFeed เป็นต้น แต่ทุกวันนี้, ผู้อ่านจะสามารถพบอุปกรณ์ต่างๆ ที่นอกเหนือไปจาก Printers ต่ออยู่ที่พอร์ตขนาน หรือ พอร์ตเครื่องพิมพ์นี้



รูปแสดง รีจิสเตอร์ต่างๆ ที่อยู่บนพอร์ตขนาน

พอร์ตขนานชนิดต่างๆ

การออกแบบเครื่องคอมพิวเตอร์ PC ได้มีวิวัฒนาการขึ้น โดยผู้ผลิตคอมพิวเตอร์หลายบริษัทได้เริ่มมีการปรับปรุงเวอร์ชันของพอร์ตขนาน (Parallel Port) ซึ่งพอร์ตชนิดใหม่นี้สามารถใช้งานได้กับพอร์ตขนานชนิดเก่า (Original design) ทุกประการ แต่พอร์ตขนานที่ออกแบบใหม่นี้จะมีความสามารถเพิ่มขึ้นโดยจะเน้นในเรื่องความเร็วในการส่งถ่ายข้อมูลเป็นหลัก และ ในเนื้อหาบทนี้จะครอบคลุมพอร์ตขนานเวอร์ชันใหม่นี้ด้วย และ ทางด้านล่างจะเป็นการสรุปโหมดชนิดต่างๆ ของพอร์ตขนาน

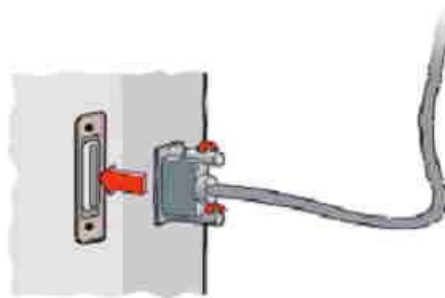
- **Original (SPP)** คือ พอร์ตขนานแบบเก่าในมาตรฐาน IBM (Original IBM PC) และ พอร์ตขนานบางเครื่องที่มีการออกแบบเหมือน Original IBM PC จะถูกเรียกว่า SPP หรือ Standard Parallel Port หรือ หลายคนรู้จักในนาม TA-type หรือ ISA-compatible นั่นเอง ซึ่งเป็นพื้นฐานในการเชื่อมต่อกับเครื่องพิมพ์ แต่ต่อมาได้เริ่มมีการเพิ่มเติมความสามารถหลายๆ อย่างเข้ามาด้วย ดังนั้น ความสามารถของพอร์ตอาจจะเพิ่มเติมจากหลักการข้างเล็กน้อย

SPPs สามารถส่งถ่ายข้อมูล 8 บิตในเวลาเดียวกันไปที่อุปกรณ์ที่ต่ออยู่กับมันได้ โดยใช้ Protocol เช่นเดียวกับการเชื่อมต่อกับเครื่องพิมพ์ แต่ SPP ไม่สามารถรับข้อมูล 8 บิตจากอุปกรณ์รอบนอกเข้ามาได้

- **SP/2-type (Simple Bidirectional)** คือ การปรับปรุงอย่างแรกของพอร์ตขนาน (Parallel Port) โดยการทำให้มันเป็น Bidirectional data port ซึ่งบริษัท IBM เป็นผู้ผลิตขึ้นโดยมีชื่อว่า PS/2 ซึ่งมันสามารถรับข้อมูล 8 บิตจากอุปกรณ์รอบนอกเข้ามายัง PC ได้ ซึ่งข้อกำหนดของ PS/2-type ที่มีมานั้นจะอ้างอิงกับพอร์ตขนาน (Parallel Port) ใดๆ ที่มี Bidirectional data port แต่จะไม่สนับสนุนการทำงานในโหมด EPP หรือ ECP ซึ่งจะได้อธิบายในหัวข้อต่อไป

- **EPP (Enhanced parallel port)** EPP ถูกพัฒนาโดยบริษัท Intel ซึ่งเป็นผู้ผลิตชิปคอมพิวเตอร์รายหนึ่ง, บนพอร์ตแบบ PS/2 บิตข้อมูลต่างๆ สามารถสื่อสารแบบ Bidirectional ได้ซึ่ง EPP ก็สามารถอ่าน หรือ เขียนข้อมูล 8 บิตได้เช่นกัน แต่การทำงานในโหมด EPP นั้นจะมีความรวดเร็วกว่า ซึ่งใช้เวลาในการอ่าน หรือ เขียนเพียง 1 cycle เท่านั้น หรือ ประมาณ 1us (บนบัส ISA) ในขณะที่พอร์ต SPP หรือ PS/2-type จะต้องใช้ถึง 4 cycle. การทำงานในโหมด EPP สามารถเปลี่ยนทิศทางการส่งข้อมูลได้อย่างรวดเร็วมาก ดังนั้น พอร์ต EPP จึงถือว่าเป็นพอร์ตที่มีประสิทธิภาพอย่างมากในการต่อใช้งานกับ Disk, Tape drives และ อุปกรณ์อื่นๆ นอกจากนั้นพอร์ต EPP ยังสามารถเลียนแบบการทำงานของ SPP หรือ SP/2-type ได้อีกด้วย

- **ECP (Extended Capabilities Port)** พอร์ต ECP ถูกเสนอขึ้นมาโดยบริษัท Hewlett Packard และ Microsoft ซึ่งมีการทำงานเหมือนกับ EPP แต่ ECP จะมี Buffer อยู่ภายใน และสามารถรองรับการส่งถ่ายข้อมูลแบบ DMA (Direct memory access) และ สามารถบีบอัดข้อมูลได้ การส่งถ่ายข้อมูลแบบ ECP นั้นมีประโยชน์กับ เครื่องพิมพ์ (Printers) , Scanners, และ อุปกรณ์รอบนอกต่างๆ ที่ต้องการการส่งถ่ายข้อมูลขนาดใหญ่ นอกจากนั้นพอร์ต ECP สามารถเลียนแบบการทำงานแบบ SPP หรือ PS/2-type หรือ EPP ได้อีกเช่นกัน



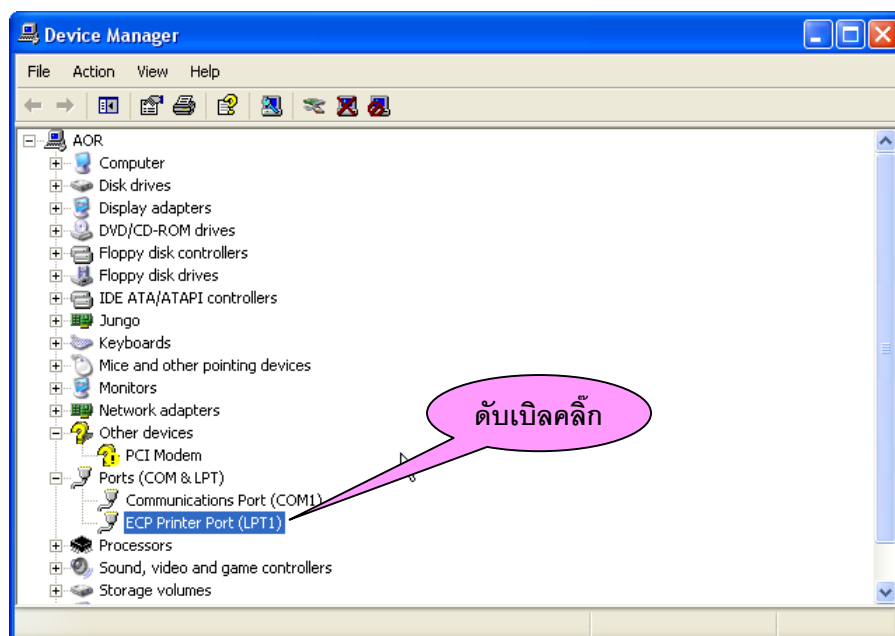
ตำแหน่งแอดเดรสของพอร์ตขนาน

โดยปกติมาตรฐานของพอร์ตขนานจะมี Base address ที่ใช้กันบ่อยๆ อยู่ 2 ค่า คือ LPT1 = 378H และ LPT2 = 278H ซึ่งผู้อ่านสามารถเปลี่ยนแปลงค่า Base address ได้ที่ BIOS

ตาราง แสดงตำแหน่งแอดเดรสของพอร์ตขนาน

ชื่อพอร์ต	LPT1	LPT 2
ชื่อรีจิสเตอร์	ฐาน16	ฐาน16
Data Register	378H	278H
Status Register	379H	279H
Control Register	37AH	27AH

ในการตรวจสอบเครื่องคอมพิวเตอร์ของผู้อ่านว่าถูกกำหนดอยู่พอร์ตไหน หรือ โหมดอะไร สามารถทำได้โดยคลิกขวาที่ไอคอน My Computer => Properties => Hardware => Device Manager จะได้ภาพที่แสดงด้านล่าง

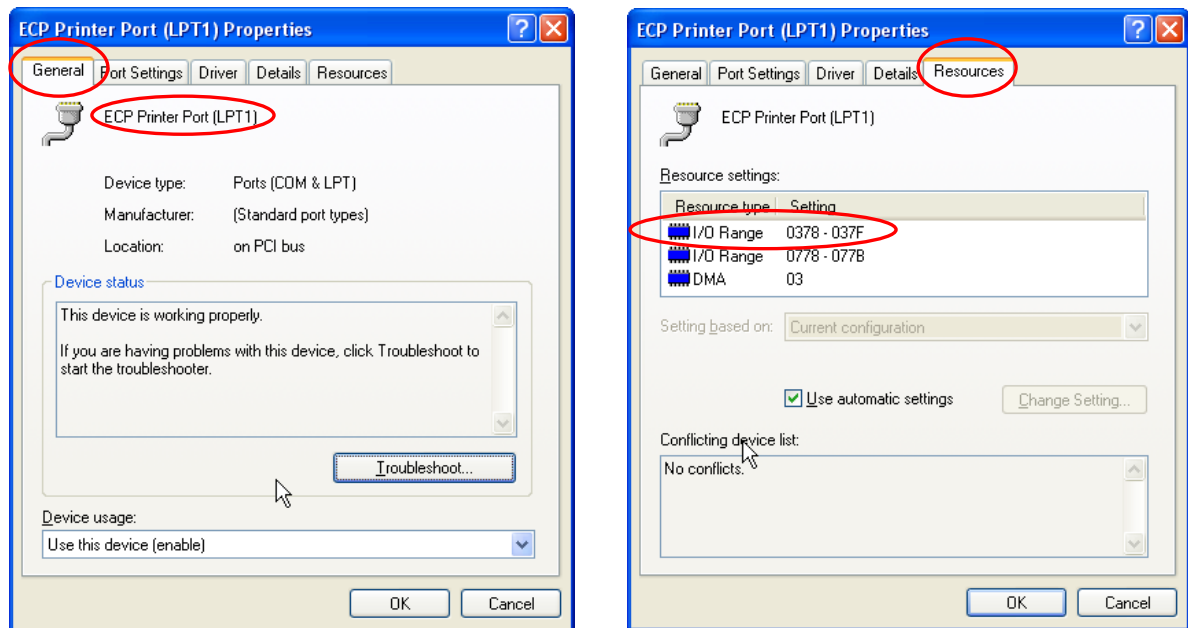


รูปแสดง การตรวจสอบพอร์ตขนานของเครื่องคอมพิวเตอร์

จากรูปทางด้านบนผู้อ่านจะพอทราบแล้วว่าคอมพิวเตอร์ถูกกำหนดอยู่ตำแหน่งใด โดยจากรูปจะเป็นการใช้งานโหมด ECP และ ใช้พอร์ต LPT1 ซึ่งหมายความว่า การเข้าถึงรีจิสเตอร์ของพอร์ตขนานจะเป็นดังนี้ คือ

- ◆ Data Register : อยู่ที่ตำแหน่งแอดเดรส 378H
- ◆ Status Register : อยู่ที่ตำแหน่งแอดเดรส 379H
- ◆ Control Register : อยู่ที่ตำแหน่งแอดเดรส 37AH

ถ้าท่านผู้อ่านยังไม่ทราบข้อมูลเหล่านี้ให้ดับเบิลคลิกตามรูปด้านล่าง ที่หัวข้อ General ผู้อ่านจะเห็นรายละเอียดทั้งหมด นอกจากนั้นในหัวข้อ Resources สามารถตรวจสอบตำแหน่งแอดเดรสได้อีกด้วย



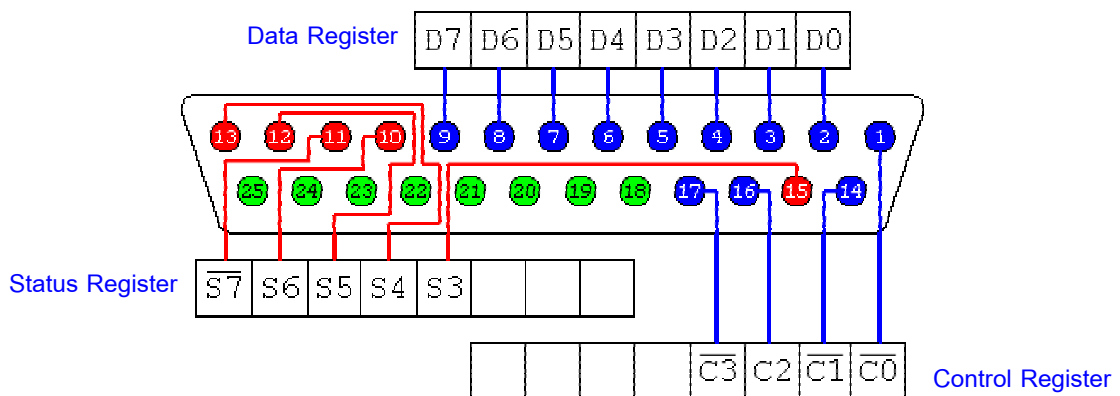
รูปแสดง การตรวจสอบพอร์ตขนานของเครื่องคอมพิวเตอร์

ขาสัญญาณบนพอร์ตนาน

ตารางด้านล่างแสดงการกำหนดชื่อขาของพอร์ตนาน (DB 25) ในโหมด EPP ซึ่งจากตารางผู้อ่านจะเห็นว่าชื่อขา และการใช้งานในแต่ละขาของโหมด SPP กับ EPP จะไม่เหมือนกัน ดังนั้น ถ้าผู้อ่านต้องการทราบรายละเอียดในโหมด SPP หรือ EPP ผู้อ่านสามารถอ่านได้จากไฟล์เอกสารใน CD-ROM ที่แถมไปกับผลิตภัณฑ์

ตาราง แสดงชื่อขาของพอร์ตนาน (D-Type 2 Pin) ในโหมด EPP

Pin	SPP Signal	EPP Signal	In/Out	Function
1	Strobe	Write	Out	A low on this line indicates a Write, High indicates a Read
2-9	Data 0-7	Data 0-7	In-Out	Data Bus. Bi-directional
10	Ack	Interrupt	In	Interrupt Line. Interrupt occurs on Positive (Rising) Edge.
11	Busy	Wait	In	Used for handshaking. A EPP cycle can be started when low, and finished when high.
12	Paper Out / End	Spare	In	Spare - Not Used in EPP Handshake
13	Select	Spare	In	Spare - Not Used in EPP Handshake
14	Auto Linefeed	Data Strobe	Out	When Low, indicates Data transfer
15	Error / Fault	Spare	In	Spare - Not used in EPP Handshake
16	Initialize	Reset	Out	Reset - Active Low
17	Select Printer	Address Strobe	Out	When low, indicates Address transfer
18-25	Ground	Ground	GND	Ground



รูป แสดงกลุ่มของขาสัญญาณ และ บิตที่มีการกลับสัญญาณ

The Data Register

Data Port หรือ Data Register, (D0 – D7) ในโหมดการทำงานแบบ SPP นั้นรีจิสเตอร์นี้จะทำหน้าที่เป็นเอาต์พุตได้อย่างเดียว แต่สำหรับโหมด EPP และ ECP จะสามารถกำหนดเป็นอินพุตได้โดยผู้อ่านจะต้องสั่งเขียน Data Register ให้เป็น '1' ทุกบิตด้วยค่า FFH ก่อนจากนั้นจึงกำหนดบิต C5 (Direction Control) ซึ่งอยู่ใน Control Register ให้เป็น '1' (Data outputs Disabled)

The Status Register

Status Port, หรือ Status Register , รีจิสเตอร์นี้มี 5 บิตที่ทำหน้าที่เป็นอินพุต คือ S3 ถึง $\overline{S7}$ โดยบิต S0 – S2 จะไม่ปรากฏที่ขาของพอร์ตขนาน และ รีจิสเตอร์นี้สามารถอ่านได้อย่างเดียวเท่านั้น ซึ่งแต่ละบิตมีหน้าที่ดังนี้ คือ

S0 : Timeout. ในโหมด EPP ถ้าบิตนี้เป็น High จะแสดงถึงการเกิด timeout ของ การส่งถ่ายข้อมูลในโหมด EPP ถ้าในโหมดอื่นบิตนี้จะไม่ใช้งาน และ บิตนี้จะไม่ปรากฏที่ขาของพอร์ตขนาน

S1 : Unused.

S2 : Unused.

S3 : nError. หรือ *nFault.* บิตนี้จะเป็น Low เมื่อเครื่องพิมพ์ตรวจพบ Error หรือ fault (อย่าสับสนกับบิต S5 (PError) นะ

S4 : SelectIn. บิตนี้จะเป็น High เมื่อเครื่องพิมพ์ On-line (เมื่อ Input data ของเครื่องพิมพ์ Enabled)

S5 : Paperend, PaperEmpty, หรือ *PError.* บิตนี้จะเป็น High เมื่อเครื่องพิมพ์กระดาษหมด

S6 : nAck. หรือ *nAcknowledge.* บิตนี้จะเป็นสัญญาณ Pulse low เมื่อเครื่องพิมพ์ได้รับข้อมูล 1 ไบต์ ถ้าได้เปิดการใช้งาน Interrupt เอาไว้จากบิต C4 จะทำให้เกิดการ Interrupt

$\overline{S7}$: Busy. บิตนี้จะเป็น Low เมื่อเครื่องพิมพ์ไม่สามารถรับข้อมูลใหม่ได้ และ บิตนี้จะกลับสัญญาณที่พอร์ตขนาน (Connector)

The Control Register

Control Port, หรือ Control Register, มี 4 บิต คือ $\overline{C0}$ จนถึง $\overline{C3}$ ในขณะที่ C4 ถึง C7 จะไม่ถูกต่อไปที่ขาของพอร์ตขนาน ในส่วนของบิตที่มีบาร์ อยู่ข้างบนมีความหมายว่าบิตๆ นั้นจะกลับสภาวะก่อนออกไปที่พอร์ตขนาน (Connector)

บิตของ Control Register เหล่านี้จะจัดวงจรเป็นแบบ Open-collector หรือ Open-Drain type ซึ่งหมายความว่าพวกมันสามารถทำหน้าที่เป็นอินพุตได้ด้วย โดยคุณผู้อ่านจะต้องเขียนลอจิก '1' ออกไปที่บิต Control ที่ผู้อ่านต้องการจะให้มันเป็นอินพุตก่อน หลังจากนั้นให้อ่านข้อมูลจากบิต Control Register นั้นๆ ได้เลย ในส่วนของหน้าที่การทำงานของบิตต่างๆ แสดงได้ดังนี้ คือ

$\overline{C0}$: *nStrobe*. บิตนี้จะใช้ในกรณีที่เครื่องพิมพ์อ่านค่าข้อมูล D0 – D7 และ โดยปกติหลังจากบิตเครื่องที่พอร์ตนานจะมีลอจิก High

$\overline{C1}$: *AutoLF*. หรือ *Automatic line feed* . ถ้าบิตนี้เป็นลอจิก Low ซึ่งจะเป็นการบอกให้เครื่องพิมพ์ทำ Line feed (ASCII code = 0AH) 1 ครั้ง หลังจากมีการกดปุ่ม Enter (Carriage Return = ASCII 0DH)

$\overline{C2}$: *nInit*. หรือ *nInitialize* ถ้าบิตนี้จ่ายสัญญาณพัลส์ลบ จะทำให้เครื่องพิมพ์นั้นรีเซ็ต และ เคลีย Buffer ซึ่งพัลส์ที่จ่ายให้กับเครื่องพิมพ์นั้นอย่างน้อยต้อง 50 ms

$\overline{C3}$: *Select*. ให้ลอจิก High เพื่อบอกให้เครื่องพิมพ์เปิดรับข้อมูลทางอินพุต

$\overline{C4}$: *Enable interrupt requests*. ถ้าให้บิตนี้เป็น High จะยอมให้ตอบสนอง Interrupt request ผ่านทางขา nAck(S6)

$\overline{C5}$: *DirectionControl*. เป็นบิตที่ใช้กำหนดทิศทางของ Data port โดยถ้าเซตเป็น '0' จะทำให้ Data port เป็น Output, ถ้ากำหนดให้เป็น '1' จะทำให้ Data port เป็น Input ซึ่งบิตนี้จะไม่ปรากฏที่ขาของพอร์ตนาน และ จะไม่สามารถใช้ร่วมกับโหมด SPP

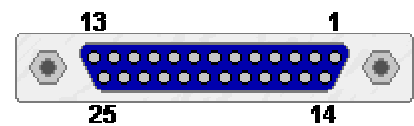
$\overline{C6}$: *Unused*.

$\overline{C7}$: *Unused*.



3

ทฤษฎีไอซี 8255



ไอซี 8255 มีขนาด 40 ขา ทำหน้าที่เป็น ไอซีขยายพอร์ตซึ่งสามารถติดต่อกับอุปกรณ์ภายนอกได้ 3 พอร์ตขนาด 8 บิต คือ PA, PB, PC โดยแต่ละพอร์ตสามารถกำหนดเป็น Input หรือ Output ได้จากการเขียนโปรแกรม ซึ่งตำแหน่งขาต่างๆ แสดงดังภาพด้านล่าง โดยเนื้อหาในบทนี้ของไอซี 8255 อ้างอิงจากเว็บไซต์ (<http://www.sharpmz.org>)

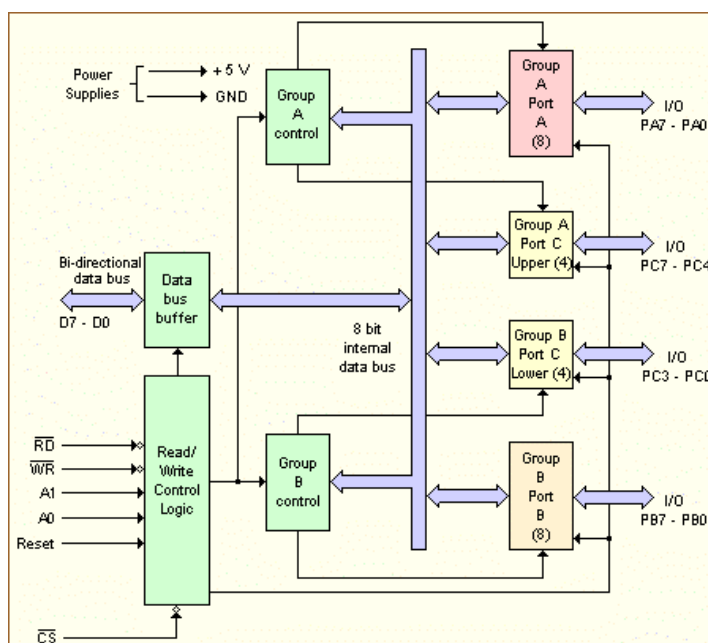
PA3	1	40	PA4
PA2	2	39	PA5
PA1	3	38	PA6
PA0	4	37	PA7
RD	5	36	WR
CS	6	35	RES ET
GND	7	34	D0
A1	8	33	D1
A0	9	32	D2
PC7	10	31	D3
PC6	11	30	D4
PC5	12	29	D5
PC4	13	28	D6
PC0	14	27	D7
PC1	15	26	VCC
PC2	16	25	PB7
PC3	17	24	PB6
PB0	18	23	PB5
PB1	19	22	PB4
PB2	20	21	PB3

รูปแสดงตำแหน่งขาของไอซี 8255

รูปทางด้านบน, แสดงตำแหน่ง และ สัญญาลักษณ์ต่างๆ ของไอซี 8255 ซึ่งฟังก์ชัน หรือ หน้าที่การทำงานของขาต่างๆ สามารถอธิบายได้ดังนี้ คือ

- $D0 - D1$ คือ ขาข้อมูลซึ่งสามารถทำหน้าที่เป็นได้ทั้ง Input/Output โดยข้อมูลที่เกิดจากการอ่านหรือ เขียน จะต้องปรากฏที่ขาเหล่านี้
- \overline{CS} (Chip Select Input) ถ้าขานี้มีสถานะ '0' ไมโครโปรเซสเซอร์ สามารถ อ่าน หรือ เขียนข้อมูลไปที่ไอซี 8255 ได้
- \overline{RD} (Read Input) เมื่อใดก็ตามที่ขา CS กับขา RD มีสถานะ '0' จะทำให้ขาเอาต์พุตของ 8255 นั้นถูก Enable ซึ่งจะส่งผลให้ข้อมูล (จาก PA, PB หรือ PC) นั้นเข้ามาอยู่บน System Data Bus ($D0 - D7$)
- \overline{WR} (Write Input) เมื่อใดก็ตามที่ขา CS กับขา WR มีสถานะ '0' จะทำให้ข้อมูลจาก System Data Bus ($D0 - D7$) ถูกเขียนออกไปที่ไอซี 8255
- $A0 - A1$ (Address Inputs) ขาสัญญาณอินพุต 2 ขานี้ทำหน้าที่เลือกพอร์ตของไอซี 8255 จาก System data bus ($D0 - D7$)
- $PA0 - PA7, PB0 - PB7$ และ $PC0 - PC7$ ขาสัญญาณเหล่านี้ใช้เป็นพอร์ต I/O ขนาด 8 บิต สามารถต่อไปที่อุปกรณ์รอบนอกต่างๆ ได้โดยตรง

โหมดการทำงานของไอซี 8255 จะมีทั้งหมด 3 โหมด คือ Mode 0, Mode 1, Mode 2 ซึ่งเนื้อหาในบทนี้ผู้เขียนจะขอกล่าวถึงเฉพาะโหมด 0 (Basic Input/Output) เท่านั้น



รูปแสดง Block diagram ของไอซี 8255

จากรูปทางด้านบนซึ่งแสดง Block diagram ของไอซี 8255 ซึ่งภายในจะแบ่งกลุ่มการทำงานออกเป็น 2 กลุ่ม คือ Group A control และ Group B control ซึ่งอธิบายได้ดังนี้ คือ

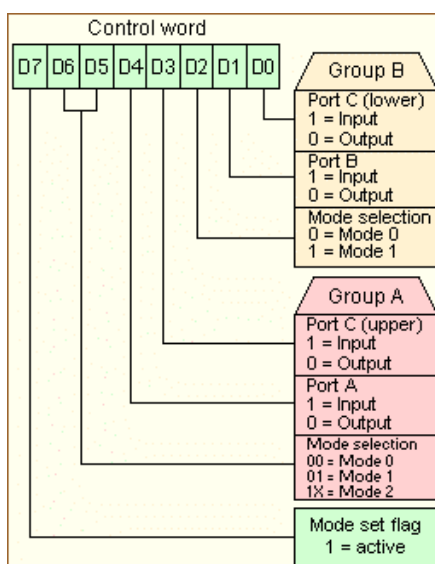
- ✿ Group A control : จะควบคุมพอร์ต A ทั้งพอร์ต และ 4 บิตบนของพอร์ต C
- ✿ Group B control : จะควบคุมพอร์ต B ทั้งพอร์ต และ 4 บิตล่างของพอร์ต C

ในส่วนของการเลือกพอร์ตนั้นผู้อ่านสามารถกำหนดได้จากขา A0 และ A1 ของไอซี 8255 ซึ่งอธิบายได้จากตารางด้านล่าง คือ

ตารางแสดงการเข้าถึงพอร์ตของไอซี 8255

A0	A1	Port
0	0	A
0	1	B
1	0	C
1	1	Control

ในการติดต่อกับไอซี 8255 ครั้งแรกผู้อ่านจะต้องส่ง Control word ไปให้กับ 8255 ก่อนเพื่อที่จะกำหนดโหมดการทำงาน รวมถึงฟังก์ชันการทำงานต่างๆ ของไอซี 8255 เสียก่อน ซึ่งการกำหนดนี้จะกำหนดเพียงครั้งเดียวเท่านั้นดังแสดงในรูปด้านล่าง คือ



รูปแสดงโครงสร้างคำสั่งของ Control Word

จากรูปทางด้านบนสามารถเขียนสรุปในรูปของตารางเพื่อง่ายต่อการนำไปใช้งานได้ดังแสดงในรูปด้านล่าง ซึ่งตารางทางด้านล่างนี้ คือ ค่า Control Word ในโหมด 0

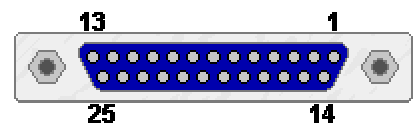
ตารางแสดงค่า Control Word ในโหมด 0 ของไอซี 8255

Contrl Word	Port A	Port B	Port C (H)	Port C (L)
80	Out	Out	Out	Out
81	Out	Out	Out	In
82	Out	In	Out	Out
83	Out	In	Out	In
88	Out	Out	In	Out
89	Out	Out	In	In
8A	Out	In	In	Out
8B	Out	In	In	In
90	In	Out	Out	Out
91	In	Out	Out	In
92	In	In	Out	Out
93	In	In	Out	In
98	In	Out	In	Out
99	In	Out	In	In
9A	In	In	In	Out
9B	In	In	In	In



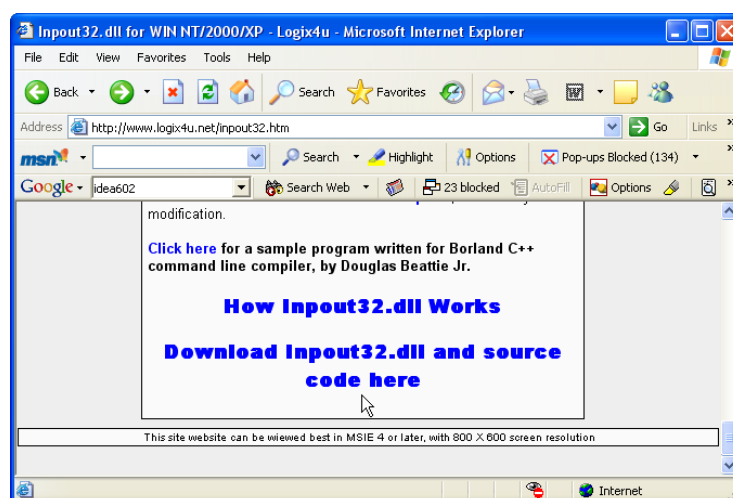
4

Inpout32.DLL ไลบรารี สำหรับติดต่อพอร์ตขนาน



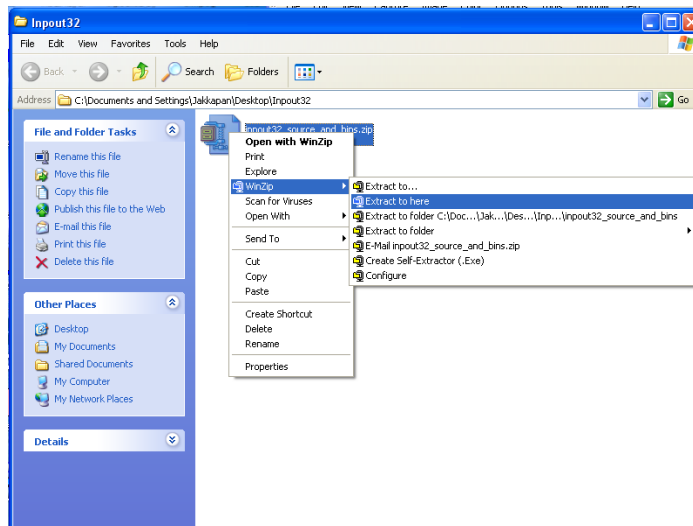
การเขียนโปรแกรมติดต่อกับพอร์ตขนาน, สำหรับ Microsoft Visual Basic 6.0 นั้นไม่ได้รับการทำงานในส่วนนี้เอาไว้ แต่อย่างไรก็ตามได้มีบริษัทหลายแห่งได้ทำฟังก์ชัน API (Application Programming Interface) สำหรับติดต่อกับพอร์ตขนานไว้ซึ่งผู้อ่านสามารถเข้าไปดาวน์โหลดมาใช้งานได้ฟรีในรูปแบบของแฟ้มข้อมูลนามสกุล “. DLL “ ซึ่งในที่นี้ผู้เขียนจะนำเสนอไฟล์ชื่อ “ Inpout32.dll “

ไฟล์ Inpout32.dll นี้สามารถใช้งานกับ Windows9x/NT/2000/XP ผู้อ่านสามารถดาวน์โหลดได้จากเว็บไซต์ <http://www.logix4u.net/inpout32.htm> จากนั้นให้คลิกเพื่อดาวน์โหลดไฟล์บนชื่อ [Download Inpout32.dll and source code here](#) ซึ่งอยู่ทางด้านล่างของ Page ดังแสดงในรูปด้านล่าง



รูปแสดงหน้าโฮมเพจของไฟล์ Inpout32.dll

จากรูปทางด้านบนเมื่อผู้อ่านทำการ Save ไฟล์แล้ว ซึ่งไฟล์ที่ได้จะเป็น Zip file จากนั้นให้คลิกขวาบนไฟล์ Zip แล้วเลือกไปที่ Winzip => Extract to here ผู้อ่านจะได้ไฟล์เดอริใหม่ชื่อ inport32_source_and_bins โดยภายในไฟล์เดอรินี้จะมีไฟล์ที่จำเป็นต่างๆ รวมทั้งข้อมูลการใช้งาน, Source code และ ตัวอย่างโปรแกรม แต่อย่างไรก็ตามผู้อ่านสามารถหาไฟล์ Inport32.dll ได้จากแผ่น CD-ROM ที่แถมไปกับผลิตภัณฑ์



รูปแสดงการ Extract ไฟล์

การติดตั้งไฟล์ Inport32.dll

การเขียนโปรแกรม VB เพื่อติดต่อกับพอร์ตขนานนั้นผู้อ่านจะต้องทำการติดตั้งไฟล์ Inport32.dll ลงในเครื่อง PC ในตำแหน่งที่ถูกต้องเสียก่อน ซึ่งไฟล์ Inport32.dll นั้นภายในได้บรรจุ API Function ที่จำเป็นสำหรับการติดต่อกับพอร์ตขนานไว้ ซึ่งทำได้ดังนี้ คือ

- คัดลอก (Copy) ไฟล์ Inport32.dll ไปไว้ที่ตำแหน่ง C:\WINDOWS\system (สำหรับ Win9x)
- คัดลอก (Copy) ไฟล์ Inport32.dll ไปไว้ที่ตำแหน่ง C:\WINDOWS\system32 (สำหรับ WinXP)

ซึ่งไฟล์ Inport32.dll นี้จะอยู่ในรูปท “ ..\inport32_source_and_bins\binaries\Dll ” (สำหรับผู้อ่านที่ดาวน์โหลดไฟล์มาจาก Internet , แต่ถ้าใช้ไฟล์ Inport32.dll จากแผ่น CD-ROM จะอยู่ในไฟล์เดอริ Dll file) และ ไฟล์นี้สามารถใช้งานได้กับ Windows9x/Me/NT/2000/XP ในส่วนของการเรียกใช้งานฟังก์ชัน API จากไฟล์ Inport32.dll จะขอออกไปอธิบายในส่วนของการอธิบายตัวอย่างโปรแกรม

ฟังก์ชันสำหรับการใช้งาน

API function ที่ผู้อ่านจะต้องใช้ในการติดต่อกับพอร์ตเครื่องพิมพ์ (Parallel Port) โดยหลักๆ จะมีอยู่ 2 ฟังก์ชันเท่านั้น คือ

INP (Reg) ‘ เมื่อ Reg คือ ตำแหน่งรีจิสเตอร์ เช่น 378H คือ Data register ของพอร์ต LPT1

OUT Reg, Data ‘ เมื่อ Reg คือ ตำแหน่งรีจิสเตอร์ และ Data คือ ข้อมูลที่ผู้อ่านต้องการส่งออกจาก PC



ภาคผนวก ก.

การเซตอัป BIOS สำหรับเลือกพอร์ต LPT

BIOS คือ กลุ่มของคำสั่งคอมพิวเตอร์ที่ถูกโหลดอยู่ในชิปหน่วยความจำ (Memory chip) บนเมนบอร์ดของเครื่องคอมพิวเตอร์ ซึ่งคำสั่งต่าง ๆ ใน BIOS จะเป็นตัวประสานการสื่อสารระหว่างระบบ Operating System (เช่น Win9X หรือ WinXP) กับ Computer Hardware โดย BIOS จะไม่ขึ้นกับระบบปฏิบัติการวินโดวส์เวอร์ชันใด ๆ นอกจากนั้นผู้อ่านสามารถ Update เวอร์ชันของ BIOS ในเวอร์ชันที่ใหม่กว่าได้จากผู้ผลิตคอมพิวเตอร์ หรือ เมนบอร์ด

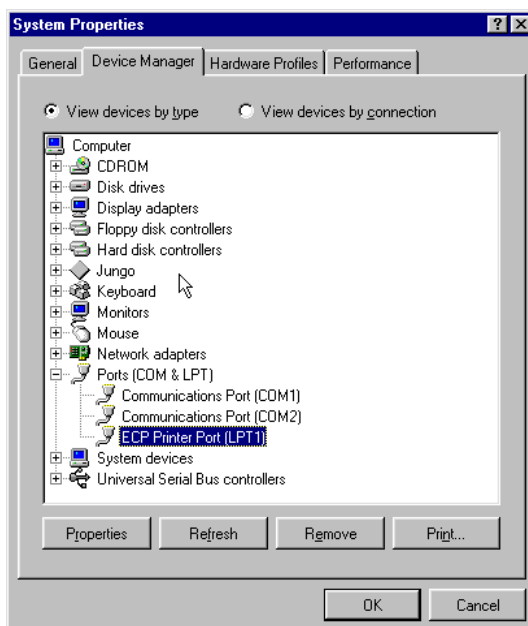
การตรวจสอบโหมดของพอร์ตขนาน

ทางเดียวในการตรวจสอบการตั้งค่า BIOS ของพอร์ตเครื่องพิมพ์ (Parallel Port) คือ การตรวจสอบคุณสมบัติใน Device Manager แต่การตรวจสอบคุณสมบัติที่ Device Manager นั้นอาจใช้ไม่ได้ 100% (เนื่องจากใน Windows NT 4.0 จะไม่มีส่วนของ Device Manager) ในการพิมพ์เอกสารโดยปกติระบบจะตั้งค่าตัวเองเป็นโหมด ECP/EPP อัตโนมัติ ซึ่งในการตรวจสอบระบบโดยดูจาก Device Manager ทำได้ดังนี้ คือ

1. เปิดหน้าต่าง Device Manager

❖ สำหรับ Windows 95, 98, และ ME

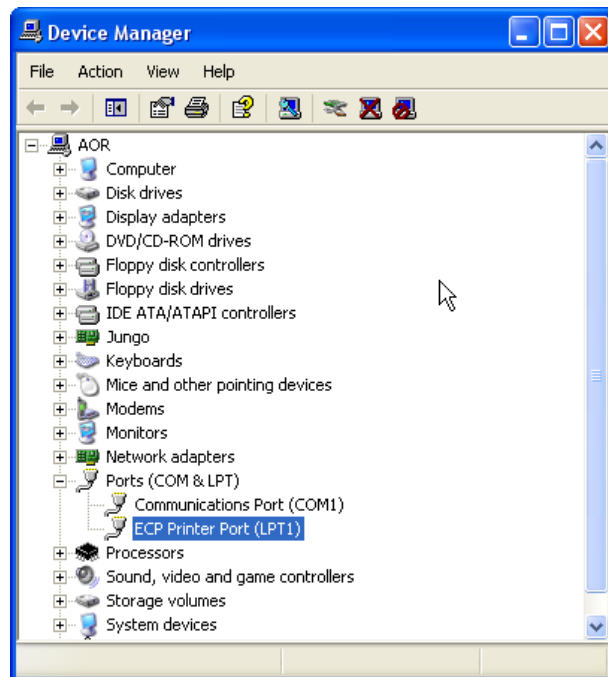
- คลิกขวาที่ My Computer จาก Desktop จากนั้นคลิกที่ Properties
- คลิกที่แท็บ Device Manager



รูปแสดงการเปิดหน้าต่าง Device Manager สำหรับ Win9x

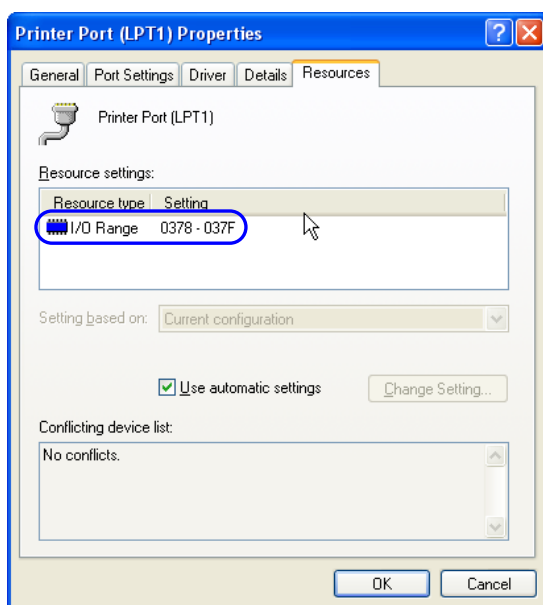
✿ สำหรับ Windows 2000 และ XP

- คลิกขวาที่ My Computer จาก Desktop จากนั้นคลิกที่ Properties (สำหรับ WinXP ให้คลิกที่เมนู Start จากนั้น คลิกขวาที่ My Computer และ ตามด้วยคลิกที่ Properties)
- คลิกที่แท็บ Hardware
- คลิกที่ปุ่ม Device Manager ในแท็บ Hardware

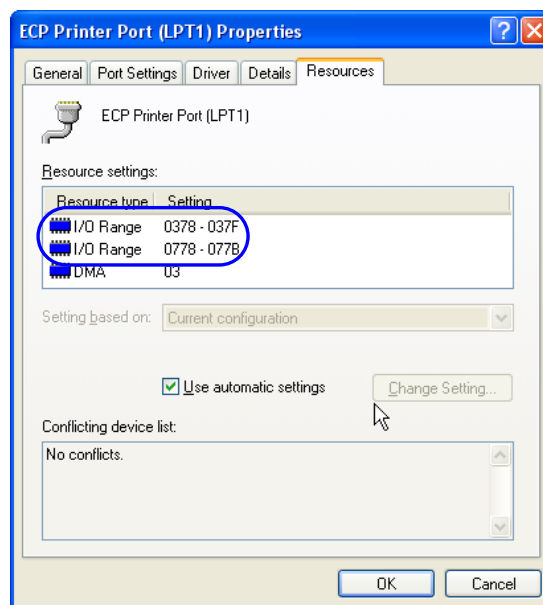


รูปแสดงการเปิดหน้าต่าง Device Manager สำหรับ Win2000/XP

2. ดับเบิลคลิกที่ Ports ในรายการที่แสดง, ซึ่งบางกรณีผู้อ่านอาจจะไม่เห็นคำว่า ECP (หรือ หมายความว่า BIOS ของผู้อ่านอาจจะถูกเซตอัปอยู่ในโหมดอื่นที่ไม่ใช่ ECP) ดังแสดงดังรูปทางด้านบน
3. ดับเบิลคลิกที่ Parallel Port (LPT1) เพื่อเข้าสู่หน้าต่าง Parallel Port Properties
4. คลิกที่แท็บ Resources, ถ้าพอร์ตเครื่องพิมพ์ (Parallel Port) อยู่ในโหมด ECP ผู้อ่านจะเห็นรายการของ I/O ranges มี 2 อัน ดังแสดงในรูปด้านล่าง แต่ถ้าอยู่ในโหมด SPP หรือ EPP รายการของ I/O ranges จะมีเพียงอันเดียว



(ก) แสดงโหมด EPP



(ข) แสดงโหมด ECP

5. ไม่ว่าจะอยู่ในโหมด EPP หรือ ECP , input/output range อันหนึ่งจะต้องเริ่มต้นที่ 0378 (สำหรับ LPT1)

Note : การทำงานของโหมดเครื่องพิมพ์อาจจะไม่ถูกต้อง, อีกทางหนึ่งในการตรวจสอบโหมดของพอร์ตเครื่องพิมพ์ คือ การตรวจสอบค่าจาก BIOS ของเครื่องคอมพิวเตอร์

การแสดงค่า และการแก้ไขค่าพอร์ตเครื่องพิมพ์ใน BIOS

วิธีการเข้าสู่ BIOS ของเครื่องคอมพิวเตอร์ต่างๆ นั้นจะขึ้นอยู่กับผู้ผลิตคอมพิวเตอร์ หรือ ผู้ผลิตเมนบอร์ดของคอมพิวเตอร์ สำหรับคอมพิวเตอร์ Laptop บางเครื่องนั้นจะมี Configuration utility สำหรับเปลี่ยนแปลงค่าต่างๆ ใน BIOS แต่โดยปกติผู้อ่านจะสามารถเข้าสู่ BIOS ได้ในขณะที่คอมพิวเตอร์กำลัง Start up ซึ่งเมื่อผู้อ่านทำการ Restart เครื่องคอมพิวเตอร์ และ เห็นข้อความ หรือ Logo ผู้ผลิตขึ้นมาจากนั้นให้ผู้อ่านกดคีย์ที่ใช้ในการเข้า BIOS

รายการคำสั่งของการเข้าโหมด BIOS ของผู้ผลิตคอมพิวเตอร์ PC ต่างๆ แสดงทางด้านล่าง ซึ่งถ้าผู้อ่านไม่สามารถเข้าโหมด BIOS ได้ ผู้อ่านควรติดต่อผู้ผลิตเครื่องคอมพิวเตอร์เพื่อขอความช่วยเหลือ หรือ รับข้อมูลเกี่ยวกับการอัปเดต BIOS

คีย์ต่างๆ ที่คุณใช้กดเพื่อเข้าสู่ BIOS

Compaq Presario

กดคีย์ F10 เมื่อโลโก Compaq สีแสงแสดงขึ้น

Dell Dimension

กดคีย์ DELETE ขณะที่ System powers up

Dell OptiPlex

กดคีย์ F2 หรือ Ctrl + Alt + Enter ในระหว่าง Startup

Dell Latitude (except LM)

กดคีย์ FN + F1 พร้อมกัน

Dell Inspiron and Latitude LM

กดคีย์ F2 ในระหว่าง Startup

E-Machines

กดคีย์ Delete ขณะที่ระบบ Starts up

Gateway

กดคีย์ F1 ขณะที่คอมพิวเตอร์ Starts

HP Pavilion (most models)

กดคีย์ F1 ขณะที่คอมพิวเตอร์ Start ที่ blue HP screen

HP Vectra and Kayak

กดคีย์ F2 ขณะที่คอมพิวเตอร์ Start ที่ blue HP screen

IBM ThinkPad

กดคีย์ FN + F1 พร้อมกัน

IBM Aptiva

กดคีย์ F1 ขณะที่คอมพิวเตอร์ Start ที่ blue IBM screen

BIOS ของ AMI, Award, และ Phoenix

มี 3 บริษัทใหญ่ๆ ที่เป็นผู้ผลิต BIOS บนคอมพิวเตอร์ คือ AMI, Award, และ Phoenix ซึ่งชื่อบริษัทมักจะแสดงตรงส่วนบนของจอภาพ BIOS (BIOS screen) ซึ่งเนื้อหาในหัวข้อนี้จะแสดงขั้นตอนการหาตำแหน่งที่ผู้อ่านจะเช็คคีย์ Parallel Port ใน BIOS ของ 3 ผู้ผลิต (AMI, Award, และ Phoenix)

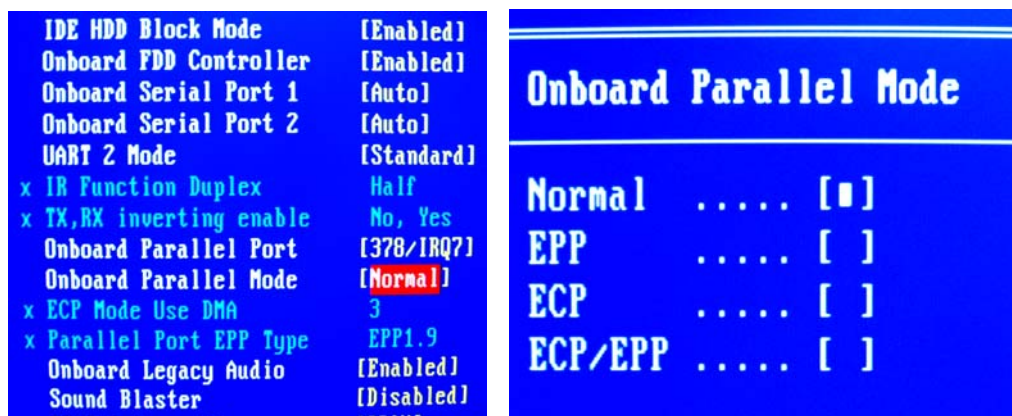
🌟 Award BIOS :

ขั้นที่ 1. จาก Main menu, ให้ผู้อ่านไปที่ Integrated Peripherals และ กด Enter ตามลำดับ



รูปแสดงหน้าต่าง Main menu

ขั้นที่ 2. กด Enter ที่หัวข้อ Onboard Parallel Mode จากนั้นเลือกโหมดที่ต้องการโดยใช้คีย์ ↑↓ และ กด Enter เพื่อตกลง

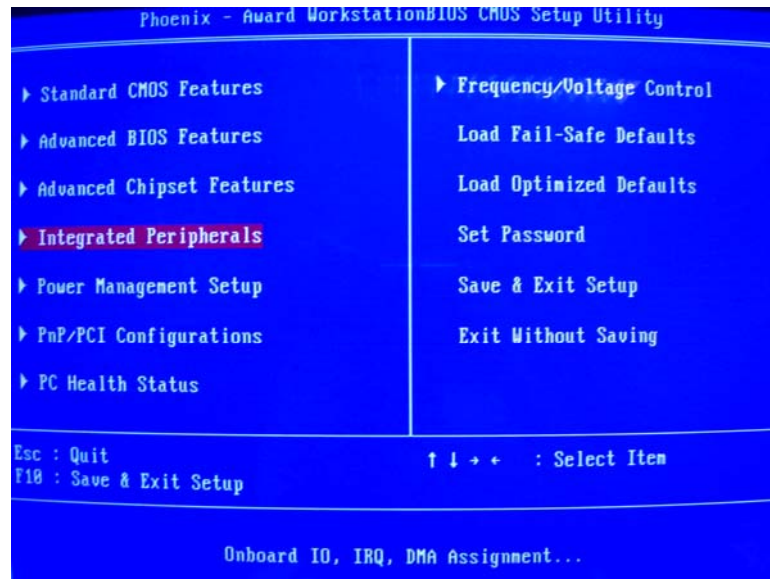


รูปแสดงการเลือกโหมดพอร์ตเครื่องพิมพ์

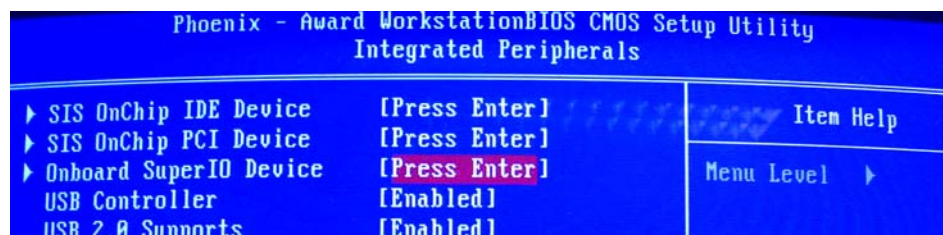
ขั้นที่ 3. กดปุ่ม Esc เพื่อออกไปที่หน้าต่าง Main menu จากนั้นทำการ Save ค่าที่ได้เปลี่ยนแปลง โดยไปที่ Save & Exit Setup

☀ Phoenix - Award :

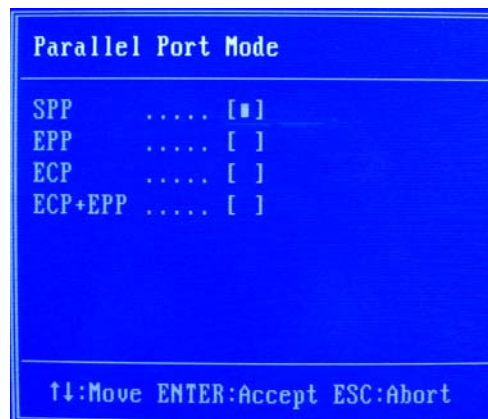
ขั้นที่ 1. จาก Main menu, ให้ผู้อ่านไปที่ Integrated Peripherals และ กด Enter ตามลำดับ



ขั้นที่ 2. กด Enter ที่หัวข้อ Onboard Super IO Device ซึ่งจะได้ดังรูปด้านล่าง



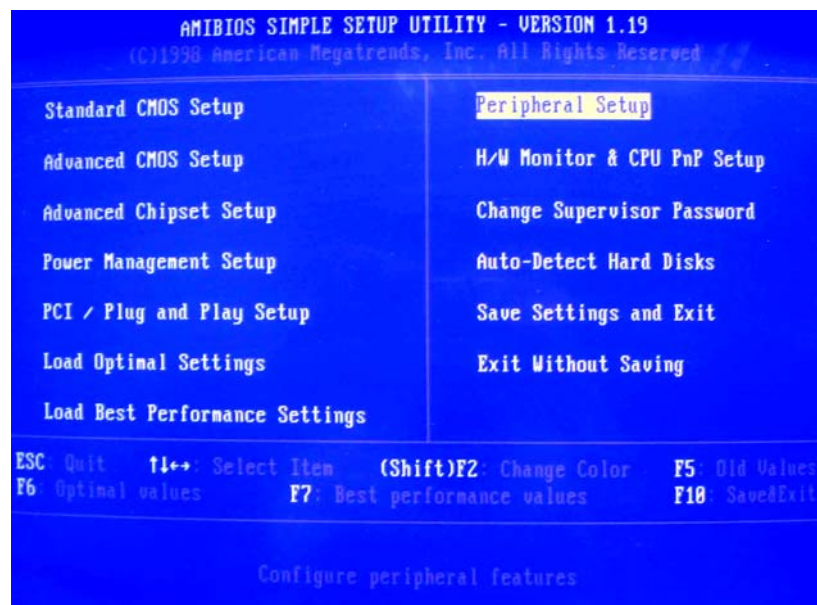
ขั้นที่ 3. ที่หัวข้อ Parallel Port Mode เป็นจุดที่ใช้กำหนดโหมดการทำงานของเครื่องพิมพ์ (Parallel Port) ซึ่งการกำหนดโหมดวิธีแรก คือ การใช้คีย์ Page Up กับ Page Down ในการเปลี่ยนโหมด หรือ อีกวิธีหนึ่ง คือ กดปุ่ม Enter ที่หัวข้อ Onboard Super IO Device ซึ่งจะมีหน้าต่างการเลือกโหมดขึ้นมาดังแสดงในรูปด้านล่าง



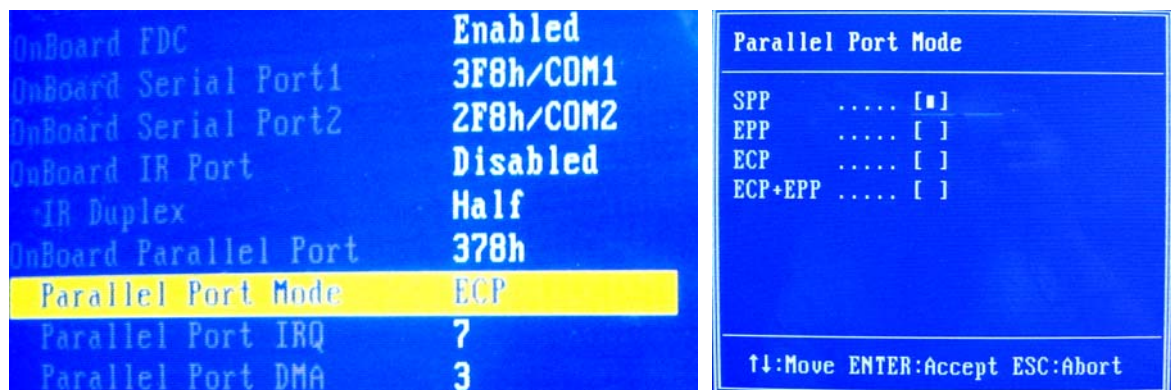
ขั้นที่ 4. เมื่อเลือกโหมดตามต้องการแล้ว ให้กดปุ่ม ESC เพื่อออกไปที่ Main menu เพื่อทำการ Save การเปลี่ยนแปลงนี้ ที่ Save & Exit Setup

☀ AMI BIOS :

ขั้นที่ 1. จาก Main menu, ให้ผู้อ่านไปที่ Peripheral Setup และ กด Enter ตามลำดับ



ขั้นที่ 2. กด Enter ที่หัวข้อ Parallel Port Mode จะได้ดังภาพด้านล่าง จากนั้นเลือกโหมดตามต้องการ



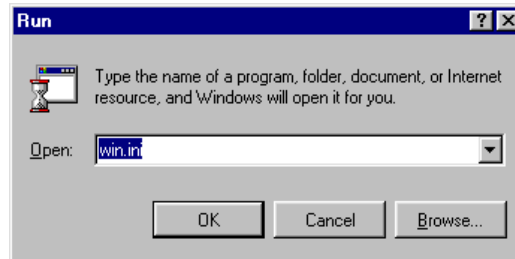
ขั้นที่ 3. เมื่อเลือกโหมดตามต้องการแล้ว ให้กดปุ่ม ESC เพื่อออกไปที่ Main menu เพื่อทำการ Save การเปลี่ยนแปลงนี้ ที่ Save Settings and Exit

ภาคผนวก ข.

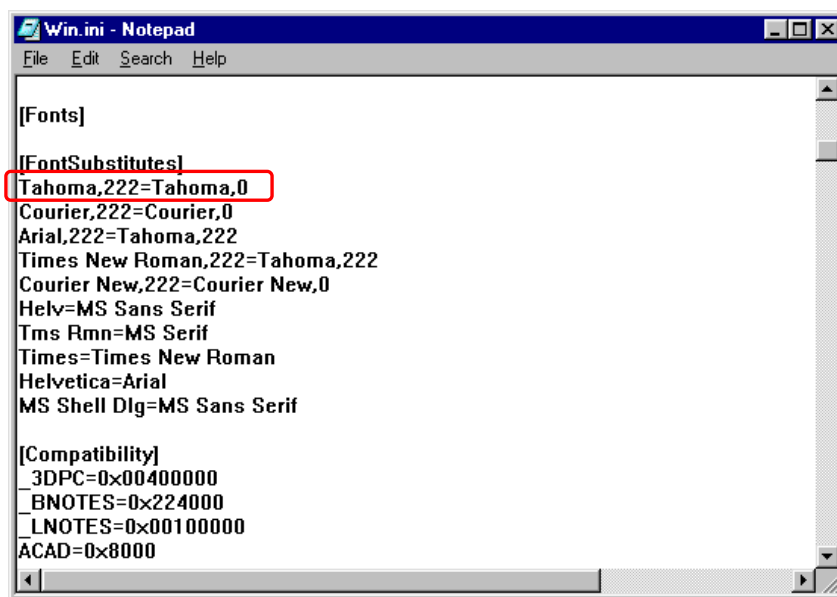
การเชื่อมต่อค่าต่างๆ ก่อนการใช้งาน VB 6.0

การแก้ไขปัญหาเกี่ยวกับฟอนต์ภาษาไทย

ขั้นที่ 1. คลิกที่ปุ่ม Start => Run แล้วพิมพ์คำว่า WIN.INI



ขั้นที่ 2. ผู้อ่านจะเห็นไฟล์ Notepad ชื่อ “ Win.ini “ ขึ้นมา จากนั้นให้เพิ่มบรรทัด Tahoma,222=Tahoma,0 หลังข้อความ [FontSubstitutes] ดังรูปด้านล่าง

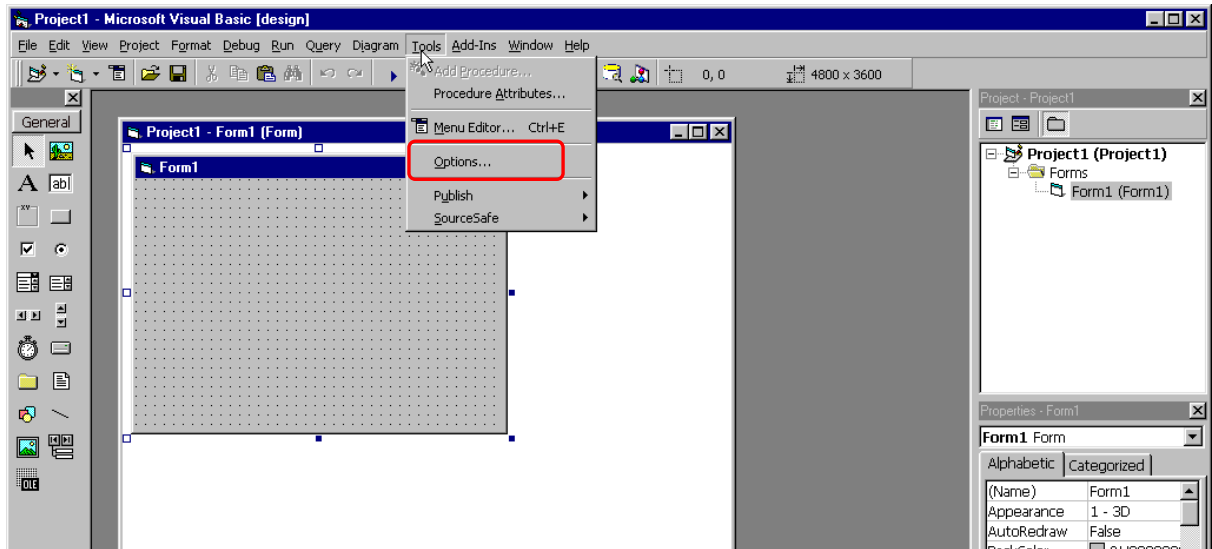


ขั้นที่ 3. ให้ผู้อ่าน Save ไฟล์ที่แก้ไขนี้โดยไปที่ File => Save จากนั้นให้ Restart คอมพิวเตอร์

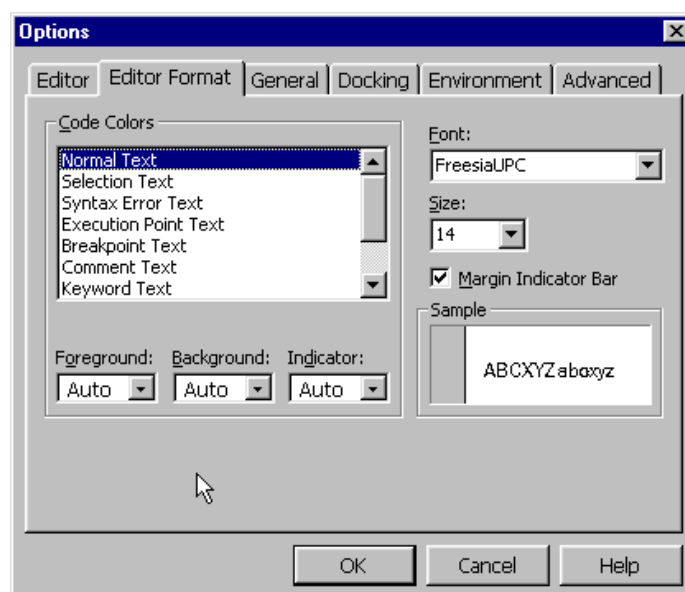
เปลี่ยนฟอนต์ที่ Code Editor ให้พิมพ์ภาษาไทยได้

การใช้งาน Visual Basic 6.0 ผู้อ่านจะเห็นว่าไม่สามารถพิมพ์ภาษาไทยในส่วนของ Code Editor ได้ ดังนั้น ขั้นตอนต่อไปนี้จะแสดงการแก้ไขฟอนต์ให้สามารถพิมพ์ภาษาไทย ได้

ขั้นที่ 1. เปิดโปรแกรม Visual Basic 6.0 ตามขั้นตอนที่ผู้อ่านเคยทำมา จากนั้นไปที่เมนู Tools => Option



ขั้นที่ 2. ที่แท็บ Editor Format ให้เลือก Font เป็น FreesiaUPC และ เลือก Size เป็น 14 จากนั้นกดปุ่ม OK ดังภาพด้านล่าง

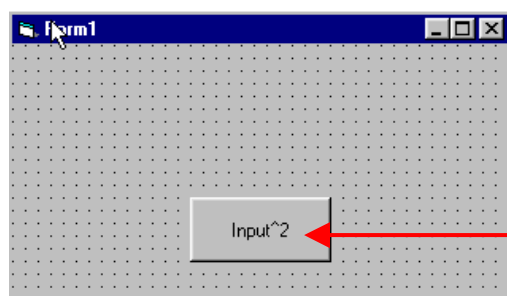


การกำหนดคำสั่ง Option Explicit

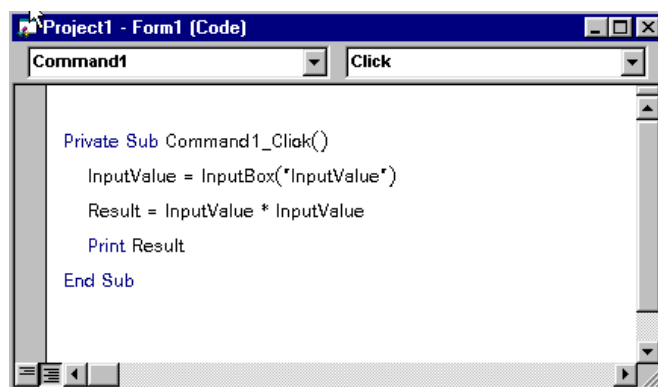
การประกาศตัวแปรในการเขียนโปรแกรมจะเป็นการบอกให้เครื่องคอมพิวเตอร์เตรียมพื้นที่ในหน่วยความจำสำหรับเก็บข้อมูลชนิดนั้นๆ ไว้ ซึ่งใน Visual Basic จะใช้คำว่า Dim (Dimension) ในการประกาศตัวแปร

สำหรับการอ้างชื่อตัวแปรใหม่ใน VB ตัวแปรนั้นจะถูกสร้างโดยอัตโนมัติ โดยผู้อ่านไม่ต้องใช้คำสั่ง Dim เลย ซึ่งการกระทำดังกล่าวมีข้อเสียตรงที่ว่า ถ้าผู้อ่านเขียนชื่อตัวแปรผิด โปรแกรมจะมองว่าเป็นการประกาศชื่อตัวแปรใหม่โดยทันที ดังตัวอย่างที่แสดงด้านล่าง

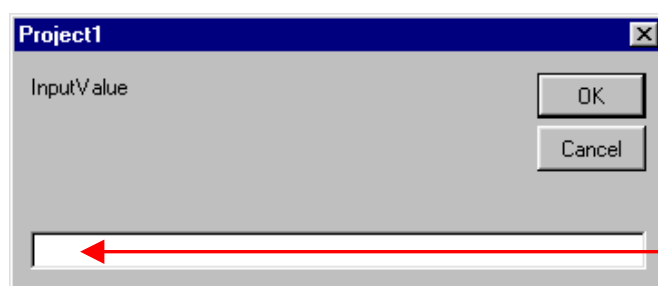
Example โปรแกรมคำนวณเลขยกกำลังสอง



Command Button ชื่อ Command1

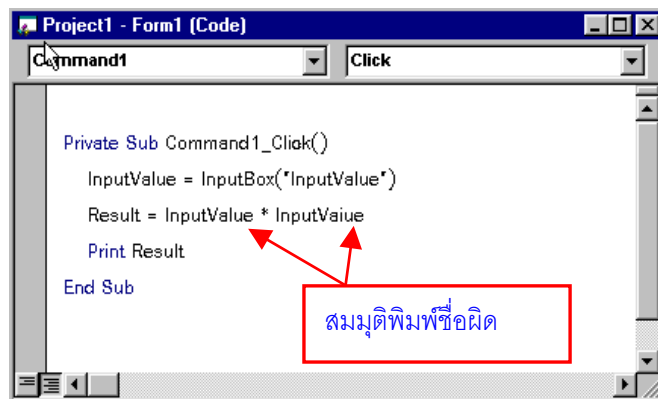


1. จากโปรแกรมทางด้านข้าง, เมื่อผู้อ่าน Run โปรแกรมผู้อ่านจะเห็นว่าโปรแกรมทำงานได้ทันทีโดยที่ไม่เกิด Error ใดๆ ขึ้นเลย ถึงแม้ว่าจะไม่ได้ประกาศตัวแปรใดๆ ด้วยคำสั่ง Dim ก็ตาม
2. ทดลองใส่ค่าจำนวนเต็ม "5" ใน InputBox ที่ปรากฏขึ้นหลังจากกดปุ่ม Input^2 ซึ่งผู้อ่านจะได้คำตอบเท่ากับ 25 ซึ่งถูกต้อง



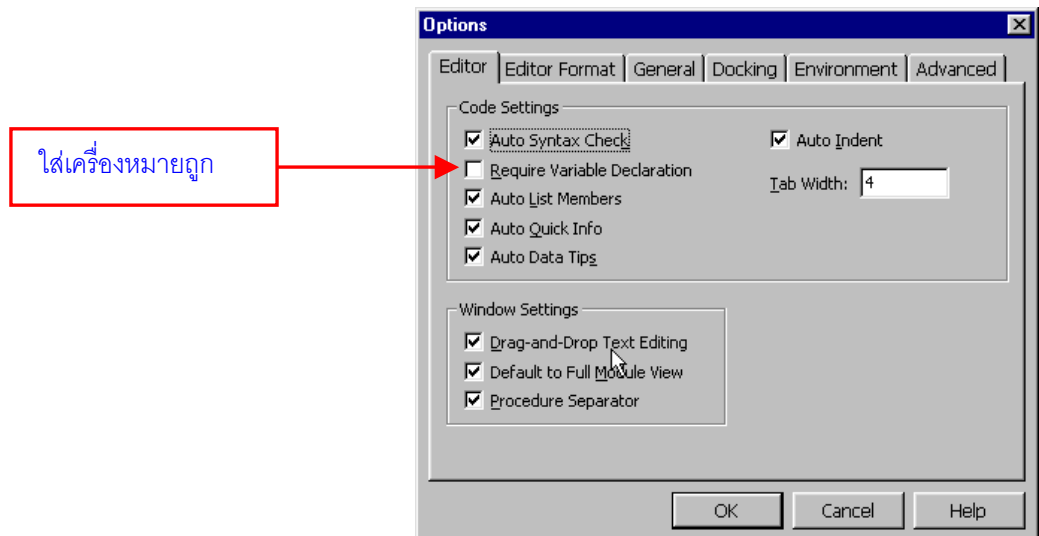
ใส่ค่าจำนวนเต็ม = 5

จากนั้นให้อ่านทดลองเปลี่ยนชื่อตัวแปรจาก InputValue เป็น InputVaiue ดังแสดงในภาพด้านล่าง

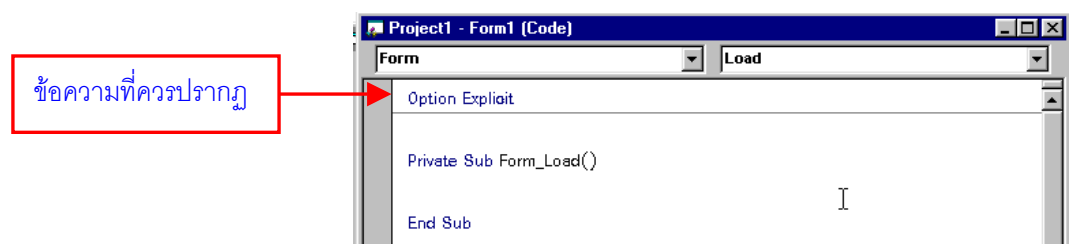


ถ้าบังเอิญพิมพ์ชื่อตัวแปรผิด ซึ่งผลจะทำให้คำตอบที่ควรจะเป็นนั้นผิดตามไปด้วยโดย Visual basic จะไม่มีการฟ้องความผิดพลาดใดๆ ทั้งสิ้น แต่จะกลับมองว่าเป็นการประกาศตัวแปรใหม่แทน ซึ่งถ้าโปรแกรมของผู้อ่านมีขนาดใหญ่มากๆ, ผู้อ่านอาจจะหาความผิดพลาดนี้ไม่เจอได้

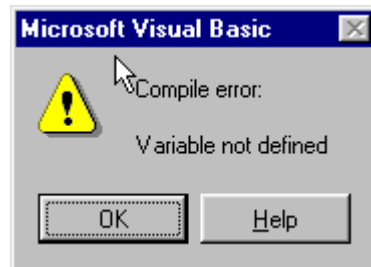
การแก้ไขปัญหานี้สามารถทำได้โดยใช้คำสั่ง Option Explicit เพื่อเป็นการบอก Visual Basic ว่าให้มีการประกาศชื่อตัวแปรก่อนการใช้งานทุกครั้ง และ เราสามารถบอกให้ Visual Basic ใส่คำสั่งนี้ให้เราทุกครั้งโดยให้อ่านไปที่เมนู Tools => Option, ในแท็บ Editor ให้อ่านติ๊กเครื่องหมายถูกที่ Require Variable Declaration จากนั้นให้กดปุ่ม OK



ทำการปิดโปรแกรม Visual Basic แล้ว เปิดโปรแกรมขึ้นมาใหม่ ทดลองดับเบิลคลิกที่ Form ผู้อ่านควรจะเห็นข้อความ Option Explicit ขึ้นมา

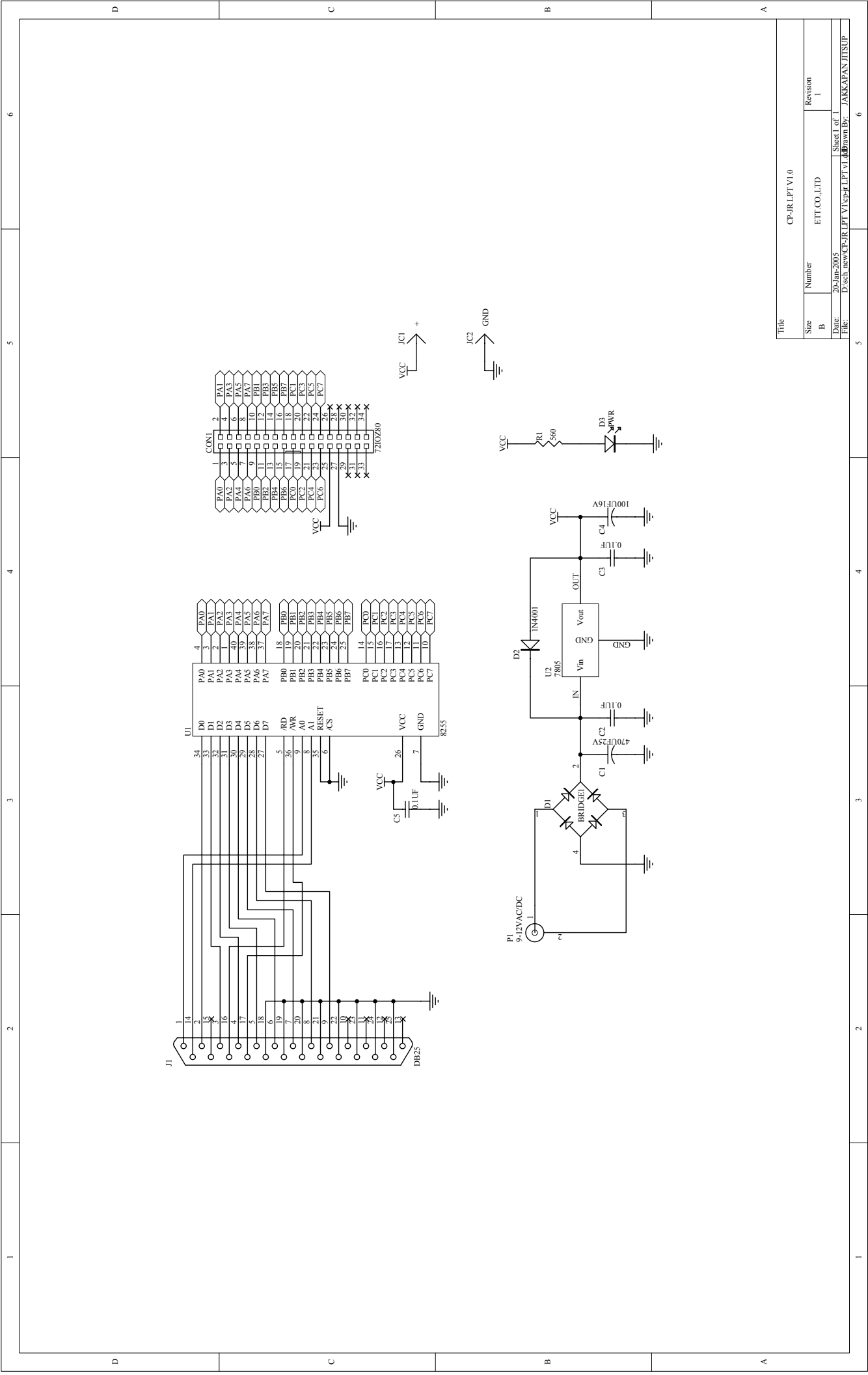


ที่นี่ถ้าผู้อ่านเผลอเขียนชื่อตัวแปรผิด หรือ ใช้ตัวแปรโดยไม่ได้ประกาศไว้ก่อนผู้อ่านจะได้รับข้อความ ดังแสดงในรูปด้านล่างจาก Visual Basic พร้อมทั้งบอกตำแหน่งที่ผิดด้วย



ภาคผนวก ค.

วงจร Schematic บอร์ด CP-JR LPT/ET-BASE LPT V1.0



Title				CP-JR LPT V1.0	
Size	Number	Revision			
B		ETT.CO.LTD		1	
Date	20-Jun-2005	Sheet 1 of 1			
File	D:\sch new\CP-JR LPT V1\sp-jr-LPT.vi	JAKKAPAN JHSUP			

*Thank you for
your order*

ett.