

เครื่องพริ้นเตอร์หยอดเหรียญ

COINBOX PRINTER



โดย

นายจักรกริช ปานเรือนแสน รหัสนักศึกษา 46015661

นายสิทธิพร เกิดเจริญ รหัสนักศึกษา 46015689

เลขหมู่.....
เลขทะเบียน..... 62599
วัน,เดือน,ปี..... 21 ส.ค. 2549

b..... 11b2b914
i.....

ปฏิญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรบัณฑิต

ภาควิชาวิศวกรรมสารสนเทศ

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2548

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญานิพนธ์ เครื่องปริ้นเตอร์หยอดเหรียญ
ชื่อนักศึกษา นายจักรกริช ปานเรือนแสน รหัสนักศึกษา 46015661
นายสิทธิพร เกิดเจริญ รหัสนักศึกษา 46015689
อาจารย์ที่ปรึกษา ผศ. อุทัย ศรีธีระวิโรจน์
อ. สรพงษ์ วชิรรัตนพรกุล
ระดับการศึกษา ปริญญาตรี วิศวกรรมศาสตรบัณฑิต
สาขาวิศวกรรมสารสนเทศ
ภาควิชา วิศวกรรมสารสนเทศ
ปีการศึกษา 2548

ปริญญานิพนธ์ฉบับนี้ ได้รับการอนุมัติให้เป็นส่วนหนึ่งของการศึกษา ตามหลักสูตร
วิศวกรรมศาสตรบัณฑิต คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหาร
ลาดกระบัง

(ผศ.อุทัย ศรีธีระวิโรจน์)

อาจารย์ที่ปรึกษา

(อาจารย์สรพงษ์ วชิรรัตนพรกุล)

อาจารย์ที่ปรึกษา

ลิขสิทธิ์ของคณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญานิพนธ์	เครื่องพรีนเตอร์หยอดเหรียญ
ชื่อนักศึกษา	นายจักรกริช ปานเรือนแสน รหัสนักศึกษา 46015661 นายสิทธิพร เกิดเจริญ รหัสนักศึกษา 46015689
อาจารย์ที่ปรึกษา	ผศ. อุทัย ศรีธีระวิโรจน์ อ. สรพงษ์ วชิรรัตนพรกุล
ระดับการศึกษา	ปริญญาตรี วิศวกรรมศาสตรบัณฑิต สาขาวิศวกรรมสารสนเทศ
ภาควิชา	วิศวกรรมสารสนเทศ
ปีการศึกษา	2548

บทคัดย่อ

โครงการนี้เป็นการนำเอา เครื่องคอมพิวเตอร์ เครื่องพรีนเตอร์ และเครื่องหยอดเหรียญมา ประยุกต์รวมกันโดยมีการทำงานแบ่งเป็น 2 ส่วนคือ ฮาร์ดแวร์ กับซอฟต์แวร์ โดยฮาร์ดแวร์จะใช้ เครื่องหยอดเหรียญในการตรวจรับเหรียญ ส่วนซอฟต์แวร์จะใช้โปรแกรมภาษาระดับสูงในการ ควบคุมฮาร์ดแวร์ เมื่อต้องการที่จะพิมพ์งานโปรแกรมจะคำนวณราคา และแสดงออกทางหน้าจอ เพื่อให้ผู้ใช้หยอดเหรียญจากนั้น โปรแกรมจะไปสั่งให้เครื่องพรีนเตอร์ทำงาน และจะทำการเก็บ ข้อมูลในการพิมพ์เป็นวันที่, เวลา, ราคา และจำนวนกระดาษที่ใช้ไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Thesis Title **Coinbox Printer**
Student Mr. Chukkrich Panreunsan ID.46015661
 Mr. Sittipron Kirdjaruan ID.46015689
Advisor Asst.Prof. Uthai Sritecraviroj
 Mr. Sorapong wachirarattanapornkul
Graduate Level Bachelor Degree of Information Engineering
Department Information Engineering
Academic 2005

Abstract

This project use computer printer and coin box in the apply to combine. The work to share two part to be a hardware and software. The hardware to use coinbox to examine catches the coin. The soft ware to use program advance language in the control hardware. User want to print, program is the calculate price will display on screen in order to user drop the coin. The program will to order the printer start it works. The print is a memory it a data in the print is a date, time, price and paper quantity to use.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ขอขอบพระคุณอาจารย์ที่ปรึกษา ผศ. อุทัย ศรีธีระวิโรจน์ และ อ. สรพงษ์ วชิรรัตนพรกุล เป็นอย่างสูงที่คอยให้คำแนะนำ คำปรึกษาอย่างเต็มที่ ตลอดจนการทำปฏิญญานิพนธ์เครื่องพรีนเตอร์ หยอดเหรียญ

ขอขอบพระคุณ คุณพ่อ คุณแม่ และอาจารย์ทุกท่านที่ให้ความสนับสนุน และกำลังใจยาม เกิดปัญหาจนกระทั่งปฏิญญานิพนธ์นี้เสร็จสมบูรณ์เป็นรูปเล่มได้

นายจักรกริช ปานเรือนแสน

นายสิทธิพร เกิดเจริญ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทคัดย่อ	ก
Abstract	ข
กิตติกรรมประกาศ	ค
สารบัญ	ง
สารบัญรูป	จ
สารบัญตาราง	ฉ
บทที่ 1 บทนำ	1
1.1 แนวคิดและที่มาของปัญหา	1
1.2 จุดประสงค์	1
1.3 ขอบเขตของโครงการ	2
1.4 ขั้นตอนการดำเนินงาน	2
บทที่ 2 ทฤษฎีที่เกี่ยวข้อง	3
2.1 การอินเตอร์เฟซกับพอร์ตอนุกรม (Serial Port Interface)	3
2.2 พอร์ตขนาน (Parallel Port)	6
2.3 เครื่องพรีนเตอร์	11
2.4 ไมโครคอนโทรลเลอร์เบสิกสเตมปี 2พี (2P)	14
2.5 Visual Basic	18
2.6 Microsoft Access 2002	19
บทที่ 3 ขั้นตอนและวิธีการดำเนินงาน	20
3.1 การออกแบบโดยรวม	21
3.2 การออกแบบแต่ละส่วน	22
3.2.1 การออกแบบโปรแกรมควบคุมไมโครคอนโทรลเลอร์	22
3.2.2 การออกแบบการติดต่อระหว่าง Visual Basic กับ Microsoft Access	25
3.2.3 การออกแบบโปรแกรมควบคุมการรับ ส่งข้อมูล และสั่งพิมพ์	37
3.2.4 การนำข้อมูลมาทำกราฟ	38

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

	หน้า
3.3 การออกแบบเพิ่มเติม	38
3.3.1 การออกแบบกล่อง	38
3.3.2 การออกแบบแหล่งจ่ายไฟ 5 โวลต์	40
3.3.3 การติดต่อเพื่อใช้งานขาพอร์ตของเบสิกแอสมบลี 2P	41
3.4 การเขียนโปรแกรมติดต่อ และควบคุมพอร์ตอนุกรม กับ Visual Basic	42
บทที่ 4 ผลการทดลอง	45
4.1 เปิดงานที่จะพิมพ์	45
4.2 ผลการหยุดเหรียญ	45
4.3 สั่งพิมพ์ที่โปรแกรม	46
4.4 เปิดงานที่จะพิมพ์ แล้วทำการสั่งพิมพ์อีกครั้ง	47
4.5 ส่งจำนวนเงิน วันที่ และเวลา ไปเก็บในฐานข้อมูล	48
4.6 การค้นหาข้อมูลรวมจากฐานข้อมูล	49
4.7 ค้นหารายได้ที่ได้รับ	50
4.8 แสดงกราฟรายได้แต่ละเดือน	51
บทที่ 5 สรุปผลของโครงการ	52
5.1 บทนำ	52
5.2 สรุปผลของโครงการ	52
5.3 แนวทางการพัฒนาต่อ	52
บรรณานุกรม	
ภาคผนวก	
ก ภาพชิ้นงาน	54
ข วงจร	58
ค Data Sheet	61
ง โปรแกรม	70
จ คู่มือการใช้งาน	88

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

	หน้า	
รูปที่ 2.1	ลักษณะของหัวข้อ	4
รูปที่ 2.2	ขาของพอร์ตขนาน	10
รูปที่ 2.3	แสดงไดอะแกรมเบื้องต้นของเบสิกแอสเอ็มปี 2P	14
รูปที่ 2.4	แสดงการจัดขาและการทำงานในแต่ละขาของเบสิกแอสเอ็มปี 2P	17
รูปที่ 3.1	เครื่องพรีนเตอร์หยอดเหรียญ	20
รูปที่ 3.2	การออกแบบโดยรวม	21
รูปที่ 3.3	การควบคุมไมโครคอนโทรลเลอร์	22
รูปที่ 3.4	ตั้งพรีนเตอร์ทำงาน	23
รูปที่ 3.5	สัญญาณพัลส์	23
รูปที่ 3.6	ไมโครคอนโทรลเลอร์ส่งข้อมูลให้กับคอมพิวเตอร์	24
รูปที่ 3.7	คอมพิวเตอร์ส่งข้อมูลให้กับไมโครคอนโทรลเลอร์	24
รูปที่ 3.8	การติดต่อระหว่างฐานข้อมูลกับ Visual Basic	25
รูปที่ 3.9	ขั้นตอนการสร้างฐานข้อมูล	26
รูปที่ 3.10	ตารางที่ได้ออกมา	26
รูปที่ 3.11	กำหนดชื่อฟิลด์ใน ตาราง 1	27
รูปที่ 3.12	กำหนดชื่อฟิลด์ใน ตาราง 2	27
รูปที่ 3.13	กำหนดชื่อฟิลด์ใน ตาราง 3	28
รูปที่ 3.14	การออกแบบการบันทึกข้อมูลของผู้ที่มาใช้บริการรวม	29
รูปที่ 3.15	การออกแบบการบันทึกข้อมูลผู้มาใช้แต่ละเดือน	30
รูปที่ 3.16	การออกแบบการบันทึกข้อมูลผู้มาใช้แต่ละปี	31
รูปที่ 3.17	การออกแบบการค้นหาข้อมูลผู้ที่มาใช้บริการ	32
รูปที่ 3.18	การออกแบบการค้นหาจำนวนเงินที่ได้รับในแต่ละวัน	33
รูปที่ 3.19	การออกแบบการค้นหาจำนวนเงินที่ได้รับในแต่ละเดือน	34
รูปที่ 3.20	การออกแบบการค้นหาจำนวนเงินที่ได้รับในแต่ละปี	35

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป(ต่อ)

	หน้า
รูปที่ 3.21 การออกแบบโปรแกรมควบคุมการรับ ส่งข้อมูลและตั้งพิมพ์	37
รูปที่ 3.22 บันทึกข้อมูล	38
รูปที่ 3.23 ตู้เครื่องหยอดเหรียญ	39
รูปที่ 3.24 ภาพด้านหน้าตู้เครื่องหยอดเหรียญ	39
รูปที่ 3.25 ชั้นวางวงจรภายในตัวกล่อง	40
รูปที่ 3.26 การออกแบบด้านหลังตู้	40
รูปที่ 3.27 วงจรแหล่งจ่ายไฟ 5 โวลต์	41
รูปที่ 3.28 การต่อสัญญาณพัลส์เข้าไมโครคอนโทรลเลอร์(BS2P)	41
รูปที่ 3.29 ทำการเพิ่มคอมโพเนนต์ MSComm	42
รูปที่ 3.30 เลือกที่รายการ Microsoft Comm Control 6.0	42
รูปที่ 3.31 คอนโทรล MSComm พร้อมทำงาน	43
รูปที่ 4.1 ตำรวจจำนวนหน้าที่ต้องการจะพิมพ์	45
รูปที่ 4.2 แสดงการหยอดเงิน	46
รูปที่ 4.3 ตั้งพิมพ์ที่โปรแกรมหยอดเหรียญ	46
รูปที่ 4.4 เปิดงานที่ต้องการพิมพ์อีกครั้งแล้วทำการตั้งพิมพ์	47
รูปที่ 4.5 นำข้อมูลไปเก็บใน Access	48
รูปที่ 4.6 การค้นหาข้อมูล	49
รูปที่ 4.7 การค้นหารายได้ที่ได้รับ	50
รูปที่ 4.8 กราฟรายได้แต่ละเดือน	51
รูปที่ ก.1 ด้านหน้าเครื่องหยอดเหรียญ	55
รูปที่ ก.2 ด้านในเครื่องหยอดเหรียญ	56
รูปที่ ก.3 วงจรเครื่องหยอดเหรียญ	57
รูปที่ ข.1 วงจรสมบูรณข์ของเบสิกแสดมปี 2p	59
รูปที่ ข.2 วงจรควบคุมแรงดัน	60

สารบัญตาราง

	หน้า
ตารางที่ 2.1 การจัดการคอนเน็คเตอร์พอร์ตอนุกรมตาม มาตรฐาน RS-232 ทั้งแบบDB-9 และDB-25	5
ตารางที่ 2.2 Data Port	7
ตารางที่ 2.3 Status Port	7
ตารางที่ 2.4 Control Port	8
ตารางที่ 2.5 ลักษณะสัญญาณของพอร์ตขนานทั้งหมด	9
ตารางที่ 2.6 สัญญาณที่ใช้ในพอร์ตขนาน	11
ตารางที่ 2.7 แสดงการทำงานในแต่ละขาของเบสิกแอสตมปี2P	18

บทที่ 1

บทนำ

1.1 แนวคิดและที่มาของปัญหา

ปัจจุบันร้านให้บริการคอมพิวเตอร์ได้จำนวนมากขึ้น ไม่ว่าจะเป็นบริการอินเทอร์เน็ต พิมพ์งาน เล่นเกมส์ หรือค้นหาหาข้อมูลต่างๆ ถึงแม้ว่าคอมพิวเตอร์จะมีราคาต่ำลงมาก แต่บ้านส่วนใหญ่ก็จะมีเครื่องพิมพ์งาน และโดยส่วนมากจะพิมพ์งานที่บ้านแล้วทำการจัดเก็บข้อมูลเพื่อนำไปพิมพ์ที่ร้านให้บริการ และในการที่จะไปพิมพ์งานที่ร้านให้บริการนั้นก็จะมีข้อจำกัด เช่นต้องว่าจ้างบุคลากรมาคอยควบคุมการพิมพ์งานจึงได้มีแนวคิดที่จะสร้างชุดเครื่องพริ้นเตอร์หยอดเหรียญขึ้นมาแก้ไขปัญหานี้ โดยสามารถนำไปใช้ในสถานที่ให้บริการชุดพริ้นเตอร์หยอดเหรียญเพื่ออำนวยความสะดวกกับบุคคลที่ต้องการจะพิมพ์งาน ส่วนผู้ให้บริการไม่จำเป็นต้องมีบุคลากรมาควบคุมงาน และยังสามารถเก็บข้อมูลของผู้ที่มาใช้บริการได้

1.2 จุดประสงค์

1. เพื่อศึกษาการควบคุมคอมพิวเตอร์กับฮาร์ดแวร์
2. เพื่อศึกษาการทำงานของพอร์ตขนาน
3. เพื่อศึกษาการเขียน โปรแกรมด้วยภาษาระดับสูง
4. เพื่อเก็บข้อมูลในการพิมพ์งาน

1.3 ขอบเขตของโครงการ

1. สามารถแปลงสัญญาณจากเครื่องหยอดเหรียญและส่งให้คอมพิวเตอร์จำนวนว่าเป็นเหรียญบาท, 5 บาท และ 10 บาท
2. สามารถใช้ภาษาระดับสูงในการควบคุมฮาร์ดแวร์
3. คอมพิวเตอร์สามารถสั่งงานให้พริ้นเตอร์พิมพ์งานได้
4. สามารถเก็บข้อมูลในการพิมพ์เป็นวันที่, เวลา, ราคา และจำนวนกระดาษที่ใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.4 ขั้นตอนการดำเนินงาน

1. ศึกษาการทำงานของเครื่องหยดเหรียญคอนโทรลเลอร์เบสิกแอสตมปี(2p) พอร์ตอนุกรมและขนานของคอมพิวเตอร์
2. ศึกษาภาษาซอฟต์แวร์ที่จะใช้ในการดำเนินโครงการ
3. ทำการออกแบบฮาร์ดแวร์และวางแผนการดำเนินงานในการพัฒนาส่วนต่างๆ
4. จัดหาอุปกรณ์ในส่วนของซอฟต์แวร์และฮาร์ดแวร์ที่จำเป็น
5. ดำเนินโครงการตามขั้นตอนที่ได้ออกแบบและวางแผนไว้

1.5 ผลที่คาดว่าจะได้รับ

1. เพื่อที่จะได้รับความรู้ในการต่ออุปกรณ์ทางด้านฮาร์ดแวร์
2. เพื่อที่จะสามารถเขียนโปรแกรมด้วยภาษาระดับสูงควบคุมฮาร์ดแวร์ได้
3. เพื่อที่จะสามารถนำเครื่องพรีนเตอร์หยดเหรียญไปให้บริการได้
4. เพื่อที่จะได้รับความรู้พื้นฐานในการทำฐานข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ทฤษฎีที่เกี่ยวข้อง

ทฤษฎีที่เกี่ยวข้องในการทำเครื่องพีซีหรือคอมพิวเตอร์นั้น สามารถแบ่งแยกออกเป็น ส่วนต่างๆ คือ

- 2.1 การอินเตอร์เฟซกับพอร์ตอนุกรม (Serial Port Interface)
- 2.2 พอร์ตขนาน (Parallel Port)
- 2.3 เครื่องพีซี
- 2.4 ไมโครคอนโทรลเลอร์เบสิกแอสตมปี 2พี (2P)
- 2.5 โปรแกรมวิซวลเบสิก 6.0 (Visual Basic 6.0)
- 2.6 Microsoft Access 2002
- 2.7 Data Access Object กับการเชื่อมต่อฐานข้อมูล

2.1 การอินเตอร์เฟซกับพอร์ตอนุกรม (Serial Port Interface)

เป็นวิธีการอินเตอร์เฟซของพีซี วิธีหนึ่งเพื่อทำการเชื่อมต่อกับอุปกรณ์อื่นๆ ซึ่งวิธีนี้จะเป็น การติดต่อแบบอนุกรมผ่านทาง พอร์ตอนุกรม ซึ่งการเชื่อมต่อแบบ พอร์ตอนุกรม นั้นจะมีความเร็ว ในการเชื่อมต่อช้ากว่า พอร์ตขนาน แต่อุปกรณ์ที่ใช้ พอร์ตขนาน จะมีราคาแพงกว่า และจะต้องใช้ สายสัญญาณมากกว่า การส่งข้อมูลทาง พอร์ตอนุกรม นั้นในเครื่องพีซีจะใช้มาตรฐาน RS-232C ซึ่งเป็นมาตรฐานการส่งข้อมูลระยะทางไม่เกิน 15 เมตร ส่วนชิปที่ใช้ในการควบคุมการทำงานนั้นนิยม ใช้ 8250 UART หรือตัวที่พัฒนามาใหม่อย่างเช่น 16550 UART ซึ่งมีการทำงานที่ดีขึ้น เราจึงต้องทำ ความเข้าใจมาตรฐานที่ใช้อยู่เพื่อนำไปประยุกต์และใช้งานตามความต้องการได้

2.1.1 ลักษณะของฮาร์ดแวร์

เพื่อที่จะให้อุปกรณ์จากผู้ผลิตต่างกันทำงานร่วมกันได้ มาตรฐานหลายชนิดได้รับการ ออกแบบขึ้นมา มาตรฐานที่ใช้กันกว้างขวางที่สุดคือ RS-232C ถูกประกาศในปี 1969 โดย Electronic Industries Association (EIA) ซึ่งเป็นการกำหนดมาตรฐานต่างๆ เช่น ลักษณะการเชื่อมต่อสัญญาณทางไฟฟ้าที่ใช้เพื่อป้องกันไม่ให้อุปกรณ์ส่งข้อมูลภายในสาย เส้นเดียวกัน อุปกรณ์สื่อสารจึงจำแนกออกเป็น 2 ประเภทคือ

- DTE (Data Terminal Equipment) คืออุปกรณ์สำหรับส่งข้อมูล เช่น คอมพิวเตอร์
- DCE (Data Communication Equipment) คืออุปกรณ์สำหรับรับข้อมูล (Input)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตามมาตรฐาน RS-232C อุปกรณ์ DTEควรใช้หัวต่อตัวผู้และอุปกรณ์ DCE ควรใช้หัวต่อตัวเมีย ซึ่งหัวต่อที่นิยมใช้กันอยู่จะเป็นชนิดD-Typeชนิด 9 ขา และ 25 ขา (บางครั้งเรียก DB-25และ DB-9)

2.1.2 คุณสมบัติทางไฟฟ้าของ Serial Port

ตามมาตรฐาน RS-232C พอสรุปได้ดังนี้

- Logic '0' หรือ "Space" มีค่า +3 Volt ถึง +25 Volt
- Logic '1' หรือ "Mark" มีค่า -3 Volt ถึง -25 Volt
- ช่วง +3 Volt ถึง -3 Volt เป็นช่วง Undefined
- Open Circuit Voltage เมื่อเทียบกับ ground ต้องไม่เกิน 25 Volt
- Short Circuit Current ต้องไม่เกิน 500 mA ซึ่ง Driver ต้องสามารถรองรับได้

2.1.3 ลักษณะของหัวต่อตามมาตรฐาน RS-232C (DB-25 & DB-9)

พอร์ตอนุกรม มีหัวต่อ 2 แบบคือ แบบ D-Type 25 Pin และแบบ D-Type 9 Pin ซึ่งทั้ง 2 แบบนี้จะเป็นชนิดตัวผู้ทางด้านของคอมพิวเตอร์ ดังนั้นจะมีอุปกรณ์ที่จะนำมาต่อกับคอมพิวเตอร์ จึงต้องมีการใช้หัวต่อชนิดตัวเมีย ซึ่งหัวต่อทั้งสองชนิดจะมีลักษณะการทำงานเหมือนกัน แต่การจัดเรียงไม่เหมือนกัน



รูปที่ 2.1 ลักษณะของหัวต่อแบบพอร์ตอนุกรมDB-9 และแบบพอร์ตขนาน DB-25

ตารางที่ 2.1 การจัดขาคอนเน็คเตอร์พอร์ตอนุกรมตามมาตรฐาน RS-232 ทั้งแบบDB-9 และDB-25

D-Type 25 Pin	D-Type 9 Pin	สัญลักษณ์	ชื่อสัญญาณ
Pin 2	Pin 3	TD	Transmit Data
Pin 3	Pin 2	RD	Receive Data
Pin 4	Pin 7	RTS	Request To Send
Pin 5	Pin 8	CTS	Clear To Send
Pin 6	Pin 6	DSR	Data Set Ready
Pin 7	Pin 5	SG	Signal Ground
Pin 8	Pin 1	CD	Carrier Detect
Pin 20	Pin 4	DTR	Data Terminal Ready
Pin 22	Pin 9	RI	Ring Indicator

2.1.4 รายละเอียดของสายสัญญาณ

สำหรับรายละเอียดของสายสัญญาณนั้นประกอบไปด้วย

- Transmit Data : TD ใช้สำหรับส่งข้อมูลอนุกรมออกจากคอมพิวเตอร์
- Receive Data : RD ใช้สำหรับรับข้อมูลอนุกรมเข้ามายังคอมพิวเตอร์
- Request To Send : RTS ใช้สำหรับส่งข้อมูลไปยังอุปกรณ์ปลายทาง เพื่อร้องขอให้อุปกรณ์ ปลายทางส่งข้อมูลกลับมา
- Clear To Send : CTS ใช้สำหรับตรวจสอบว่าอุปกรณ์ที่เชื่อมต่อด้วยพร้อมที่จะรับข้อมูล หรือไม่ โดยจะคอยรับสัญญาณ RTS เมื่อทุกอย่างพร้อมก็จะทำการส่งข้อมูลออกทางขา TD
- Data Set Ready : DSR ใช้สำหรับตรวจสอบการเชื่อมต่อกัน ระหว่างคอมพิวเตอร์กับอุปกรณ์ปลายทาง จะใช้คู่กับขา DTR
- Signal Ground : SG เป็นกราวด์ของระบบ
- Carrier Detect : CD ขานี้จะ Active เมื่อมีการส่งสัญญาณ Carrier จากโมเด็ม
- Data Terminal Ready : DTR ใช้สำหรับบอกให้อุปกรณ์ปลายทางรับรู้ว่าต้องการจะติดต่อกับ โดยขา DTR นี้ต้องเชื่อมต่อกับขา DSR ของอุปกรณ์ปลายทาง

2.2 พอร์ตขนาน

พอร์ตขนาน หรือ Parallel Port นั้นเดิมเรียกว่า Printer Port เพราะการใช้งานส่วนใหญ่กับพอร์ตขนานเป็นการใช้งานโดยต่อกับพริ้นเตอร์เป็นหลัก โดยที่พอร์ตขนานนั้นสามารถให้ความเร็วในการส่งผ่านข้อมูลได้รวดเร็วกว่าพอร์ตอนุกรมราว 8-10 เท่า ซึ่งสามารถส่งข้อมูลขนาน 8 บิตได้เลย ลักษณะหัวต่อของพอร์ตขนานจะเป็นแบบ D-type 25 pin ตัวเมียอยู่หลังเครื่องคอมพิวเตอร์ทั่วไป ปกติแล้วใจใช้ในการติดต่อกับเครื่องพริ้นเตอร์

อย่างไรก็ตามนอกจากพอร์ตขนานจะใช้ติดต่อกับเครื่องพริ้นเตอร์แล้ว ยังสามารถใช้งานเชื่อมต่อกับอุปกรณ์ชนิดอื่นๆ ได้อีกซึ่งการใช้พอร์ตขนานในงานการเชื่อมต่อนั้นถูกใช้งานกันอย่างแพร่หลายมาก ทั้งนี้เพราะสามารถรับและส่งข้อมูลในลักษณะขนานได้ทำให้นำไปใช้ในการควบคุมอุปกรณ์ภายนอกได้เป็นอย่างดี อีกทั้งลักษณะแรงดันที่จ่ายออกมาก็เป็น TTL โดยสัญญาณลอจิก “1” จะเท่ากับ 5 โวลต์และลอจิก “0” จะเท่ากับ 0 โวลต์ทำให้ง่ายในการออกแบบวงจร และการประยุกต์ใช้งาน

ข้อเสียก็คือไม่สามารถทำงานในระยะทางที่ไกลๆ ได้เพราะจะเกิดความผิดพลาดของข้อมูลได้ง่ายมากอันเนื่องจากแรงดันที่ไม่สม่ำเสมอ และจะสิ้นเปลืองค่าใช้จ่ายในเรื่องของสายเพราะต้องใช้สายสัญญาณหลายเส้น

พอร์ตขนานของเครื่องคอมพิวเตอร์ประกอบด้วยสัญญาณทั้งหมด 25 เส้นสัญญาณ แต่ใช้งานจริงๆ เพียง 17 เส้นสัญญาณ โดยสัญญาณจะแบ่งออกได้เป็น 3 กลุ่มใหญ่ๆ ตามลักษณะหน้าที่ของสัญญาณประกอบด้วย

- Data Port จำนวน 8 เส้นสัญญาณ
- Status Port จำนวน 5 เส้นสัญญาณ
- Control Port จำนวน 4 เส้นสัญญาณ

2.2.1 ลักษณะของสัญญาณ

ขาสัญญาณของพอร์ตขนานแบ่งได้เป็น 3 กลุ่มดังนี้

2.2.1.1 Data Port จะมีอยู่ 8 ขา หรือ 8 pin (ตั้งแต่ขาที่ 2 ถึงขาที่ 9) บางทีมักถูกเรียกว่า Data Register ซึ่ง Register ตัวนี้จะส่งค่าได้อย่างเดียวไม่สามารถรับค่าได้

ตารางที่ 2.2 Data Port

Name	Read / Write	Bit No.	Signal Name
Data Port	Write	Bit 7	Data 7 (pin 9)
		Bit 6	Data 6 (pin 8)
		Bit 5	Data 5 (pin 7)
		Bit 4	Data 4 (pin 6)
		Bit 3	Data 3 (pin 5)
		Bit 2	Data 2 (pin 4)
		Bit 1	Data 1 (pin 3)
		Bit 0	Data 0 (pin 2)

2.2.1.2 STATUS PORT เป็นพอร์ตที่อ่านได้อย่างเดียวไม่สามารถเขียนข้อมูลได้ พอร์ตนี้จะมีสัญญาณเข้าอยู่ 5 สัญญาณ และสัญญาณ IRQ กับสัญญาณสงวนไว้อีกสองบิต โดยสัญญาณ Busy จะ Active Low

ตารางที่ 2.3 Status Port

Name	Read / Write	Bit No.	Signal Name
Status Port	Read	Bit 7	Busy
		Bit 6	nAck
		Bit 5	PaperEnd
		Bit 4	Select
		Bit 3	nError
		Bit 2	IRQ (Not)
		Bit 1	Reserved
		Bit 0	REserved

สำหรับลักษณะการทำงานของแต่ละบิตใน Status Port

- Bit 7 Busy เมื่อ Active หมายถึงพริ้นเตอร์จะไม่รับข้อมูล
- Bit 6 nAck เมื่อ Active หมายถึงพริ้นเตอร์พร้อมที่จะทำงาน (Active Low)
- Bit 5 Paper End เมื่อ Active หมายถึงพริ้นเตอร์ไม่มีกระดาษ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Bit 4 Select	เมื่อ Active หมายถึงเลือกพริ้นเตอร์
Bit 3 nError	เมื่อ Active หมายถึงพริ้นเตอร์เกิดข้อผิดพลาด (Active Low)
Bit 2, Bit 1, Bit 0	ไม่ใช่

2.2.1.3 CONTROL PORT เป็นพอร์ตที่ใช้ในการควบคุมพริ้นเตอร์ สัญญาณในกลุ่มนี้จะ Active Low ยกเว้น สัญญาณ Intialize เท่านั้นที่ไม่ถูก Invert

ตารางที่ 2.4 Control Port

Name	Read / Write	Bit No.	Signal Name
Control Port	Read /Write	Bit 3	nESelect (pin 17)
		Bit 2	nInitialize (pin 16)
		Bit 1	nAutoFeed (pin 14)
		Bit 0	nStrobe (pin 1)

ลักษณะการทำงานของแต่ละบิตใน Control Port

Bit 3	nSelect Printer	เมื่อ Active หมายถึงเลือกพริ้นเตอร์
Bit 2	nInitialize	เมื่อ Active หมายถึงรีเซ็ตพริ้นเตอร์
Bit 1	Auto Feed	เมื่อ Active หมายถึงพริ้นเตอร์กระทำ Line Feed
Bit 0	nStrobe	เมื่อ Active หมายถึงการบอกให้พริ้นเตอร์ทราบว่าข้อมูลเข้ามาแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.2 สรุปลักษณะสัญญาณของพอร์ตขนานทั้งหมด

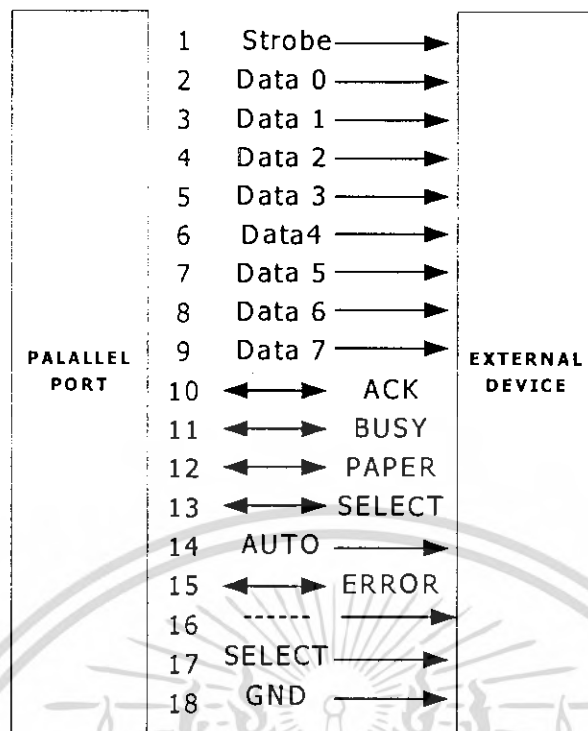
สำหรับลักษณะของสัญญาณของพอร์ตขนานทั้งหมดมีรายละเอียดดังตาราง

ตารางที่ 2.5 ลักษณะสัญญาณของพอร์ตขนานทั้งหมด

Pin NO.(D-Type25)	Signal Name	Bit	Direction (In /Out)
1	nStrobe	-Co	Output
2	Data 0 (Bit 0)	D0	Output
3	Data 1 (Bit 1)	D1	Output
4	Data 2 (Bit 2)	D2	Output
5	Data 3 (Bit 3)	D3	Output
6	Data 4 (Bit 4)	D4	Output
7	Data 5 (Bit 5)	D5	Output
8	Data 6 (Bit 6)	D6	Output
9	Data 7 (Bit7)	D7	Output
10	nAck	S6	Input
11	Busy	-S7	Input
12	PaperEnd	S5	Input
13	Select	S4	Input
14	nAutoFeed	-C1	Output
15	nError	S3	Input
16	nIntialize	C2	Output
17	nSelectPrinter	-C3	Output
18-25	Ground	-	-

2.2.3 ขาพอร์ตขนาน

สำหรับการต่ออุปกรณ์ภายนอกเข้ากับพอร์ตขนานที่ขาต่างๆ จะมีทิศทางของการรับส่งข้อมูล หรือ Pin Out ของพอร์ตขนานดังนี้



รูปที่ 2.2 ขาของพอร์ตขนาน

2.2.4 รูปแบบการติดต่อผ่านทางพอร์ตขนาน

เมื่อมีการติดต่อกับอุปกรณ์ หรือพรีนเตอร์ใดๆ ผ่านทางพอร์ตขนานนั้นการทำงานจะเริ่มจากคอมพิวเตอร์จะส่งสัญญาณข้อมูลขนาด 8 บิต ออกทาง Data Port แล้วสร้างสัญญาณ Strobe ให้เป็น Low ส่งไปยังอุปกรณ์ หรือพรีนเตอร์เพื่อบอกให้ทราบว่าข้อมูลพร้อมที่เตรียมจะส่งให้แล้วจากนั้นคอมพิวเตอร์ ก็จะรอรับการตอบกลับจากอุปกรณ์หรือพรีนเตอร์ที่ต่อกับพอร์ตขนานนั้น โดยสิ่งที่ตอบกลับมามีอยู่ 2 ลักษณะคือ

2.2.4.1 สัญญาณ nACK เพื่อเป็นการแสดงว่าอุปกรณ์ หรือพรีนเตอร์พร้อมที่จะรับสัญญาณข้อมูล โดยจะสร้างสัญญาณ Acknowledge เป็น Low ตอบกลับไป

2.2.4.2 สัญญาณ Busy เพื่อเป็นการแสดงว่าอุปกรณ์ หรือพรีนเตอร์ไม่ว่าง ไม่พร้อมรับข้อมูลจากรูปข้างต้นเราสามารถสรุปรายละเอียดเกี่ยวกับสัญญาณที่ใช้ใน พอร์ตขนานได้ดังนี้

ตารางที่ 2.6 สัญญาณที่ใช้ในพอร์ตขนาน

สัญญาณ	ผู้ส่ง	สิ่งที่ส่ง	ผู้รับ
ข้อมูล 8 บิต	คอมพิวเตอร์	ข้อมูลที่คอมพิวเตอร์ส่งไปที่พอร์ต	พอร์ตขนาน
Strobe	คอมพิวเตอร์	แจ้งให้พอร์ตทราบว่าข้อมูลชุดใหม่ส่งมา	พอร์ตขนาน
Acknowledge	พอร์ตขนาน	ตอบกลับมาว่าพร้อมรับข้อมูลแล้ว	คอมพิวเตอร์
Busy	พอร์ตขนาน	ตอบกลับมาว่าไม่พร้อม	คอมพิวเตอร์
Error	พอร์ตขนาน	แจ้งข้อผิดพลาดกลับมาให้คอมพิวเตอร์	คอมพิวเตอร์
Reset	คอมพิวเตอร์	ให้รีเซ็ตข้อมูล หรือรีเซ็ตพริ้นเตอร์	พอร์ตขนาน

2.3 เครื่องพริ้นเตอร์

เครื่องพริ้นเตอร์ (printer) เป็นอุปกรณ์ที่อยู่คู่กับเครื่องคอมพิวเตอร์มาตั้งแต่ไหนแต่ไรแล้ว เทคโนโลยีของพริ้นเตอร์ในการแปลงข้อมูลผลลัพธ์จากการทำงานของคอมพิวเตอร์ ให้กลายเป็นข้อความบนกระดาษก็ได้มีการปรับปรุงเปลี่ยนแปลงไปเรื่อย ๆ ตามยุคสมัย จากที่ใช้หัวพิมพ์เป็นตัวอักษรอักษรแต่ละตัว เช่น ในพริ้นเตอร์ของเครื่องเมนเฟรมยุคก่อน (Band printer ที่มีตัวอักษรเรียงกันอยู่บนแถบหรือ Band แล้วพิมพ์ทีละบรรทัด บางทีก็เลือกกว่าเป็น Line printer) มาจนถึงแบบที่ใช้หัวพิมพ์เป็นจุดเล็กๆ เรียงต่อกันเป็นตัวอักษรหรือรูปภาพ ที่เลือกกันว่าพริ้นเตอร์แบบตารางจุด (Dot-matrix) ซึ่งนอกจากจะพิมพ์ตัวอักษรเป็นรูปแบบต่าง ๆ ได้อย่างสวยงามขึ้นแล้ว ยังสามารถพิมพ์รูปภาพได้อีกด้วย ส่วนเครื่องพิมพ์ที่เกิดขึ้นในยุคหลังคือพวกที่เรียกว่าเป็น Non-impact printer ทั้งหลายซึ่งไม่มีการกระทบกันระหว่างหัวพิมพ์กับกระดาษ ได้แก่พวกที่ใช้เครื่องพริ้นเตอร์แสงเลเซอร์ (Laser printer) หรือพวกที่ใช้หมึกหรืออิงค์เจ็ท (Inkjet) ซึ่งนอกจากจะมีความละเอียดสูง(300 จุดต่อนิ้วขึ้นไป ในขณะที่ Dot-matrix ทำได้เพียงประมาณ 180-360 จุดต่อนิ้วเท่านั้น) แล้วยังมีความเร็วมากกว่าพริ้นเตอร์แบบ Dot-matrix อีกด้วย

2.3.1 ส่วนประกอบของพริ้นเตอร์

ก่อนจะเข้าไปถึงละเอียดเทคโนโลยีของพริ้นเตอร์แต่ละประเภท ขอให้มองที่ภาพรวมของการทำงานประสานกันระหว่างพริ้นเตอร์กับคอมพิวเตอร์ก่อน กล่าวคือการทำงานที่พริ้นเตอร์จะพิมพ์ข้อมูลในรูปแบบต่าง ๆ ออกมาได้ นั้น ซึ่พริ้นเตอร์ที่ซีพียูจะต้องส่งข้อมูลมายังพริ้นเตอร์ก่อน แล้วภายในพริ้นเตอร์จะมีอุปกรณ์ที่มีความสามารถในการแปลงข้อมูลเหล่านั้นให้เป็นรูปแบบซึ่งจะนำไปสร้างตัวอักษรหรือภาพบนกระดาษโดยกลไกการพิมพ์ของพริ้นเตอร์อีกทีหนึ่ง ระบบควบคุมนี้จะว่าไปแล้วก็คือคอมพิวเตอร์อีกเครื่องหนึ่งนั่นเอง กล่าวคือจะต้องมีซีพียูและหน่วยความจำของตัวเองเพื่อเก็บโปรแกรมตลอดจนเป็นที่พักของข้อมูล เพื่อจะประมวลผลส่งเป็นสัญญาณควบคุมไปที่กลไกการพิมพ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.2 ระบบควบคุมของพริ้นเตอร์

ระบบควบคุมของพริ้นเตอร์โดยทั่วไปจะถูกออกแบบมาให้ใช้พียูเป็นไมโครโพรเซสเซอร์ไม่ว่าจะเป็นตระกูลอินเทลเอง หรือที่นิยมกันมากกว่าก็เช่นตระกูล 68000 ของ Motorola หรือในพริ้นเตอร์ยุคใหม่ ๆ โดยเฉพาะพวกพริ้นเตอร์แบบเลเซอร์จะใช้พียูแบบ RISC เช่น i860 และ i960 ของอินเทลหรือ AMD 29000 ของ AMD เป็นต้น ทั้งนี้ขึ้นอยู่กับความซับซ้อนในลักษณะการทำงานของพริ้นเตอร์เป็นหลัก ถ้ากระบวนการทำงานมีความซับซ้อนมากก็ต้องใช้พียูที่มีความเร็วสูง ถัดจากนั้นก็จะเป็นหน่วยความจำที่ใช้เป็นบัฟเฟอร์หรือที่พักข้อมูล ซึ่งจะมีมากน้อยแตกต่างกันไปตามกลไกการพิมพ์และวิธีทำงานของพริ้นเตอร์แต่ละแบบ ตัวอย่างเช่น พริ้นเตอร์แบบ Dot-matrix พิมพ์งานทีละบรรทัดหนึ่ง (กวาดหัวพิมพ์ไปมา 1 ครั้งได้ 1 บรรทัด) อย่างน้อยก็ควรจะมีที่พักข้อมูลนี้ให้เพียงพอกับการพิมพ์บรรทัดหนึ่ง (ซึ่งอาจเป็น 80,132 หรือ 225 ตัวอักษรแล้วแต่ขนาดของพริ้นเตอร์และตัวอักษรที่ใช้) หรือพริ้นเตอร์แบบเลเซอร์ใช้วิธีการพิมพ์โดยสร้างภาพทีละ 1 หน้ากระดาษขึ้นมาก่อน ก็ต้องมีหน่วยความจำให้เพียงพอกับการเก็บข้อมูลทั้งหน้า หรือถ้ามีรูปกราฟฟิกต่าง ๆ ด้วยก็ต้องใช้หน่วยความจำมากขึ้นไปอีก

นอกจากหน่วยความจำที่ใช้ในการเก็บข้อมูลที่จะพิมพ์แล้ว พียูของพริ้นเตอร์เองก็มีหน่วยความจำสำหรับเก็บรูปแบบตัวอักษรหรือฟอนต์ (Font) และโปรแกรมที่พียูจะใช้ในการทำงานซึ่งอาจเก็บในหน่วยความจำแบบ ROM ที่อยู่ในเครื่องพริ้นเตอร์นั่นเอง หรือถ้าไม่พอก็ต้องมีอุปกรณ์เก็บข้อมูลอื่น ๆ เพิ่ม เช่น เป็นตลับ (cartridge) หรือการ์ดเก็บข้อมูลแบบฟอนต์สำหรับนำมาเสียบเพิ่มที่พริ้นเตอร์ (ซึ่งภายในก็คือ ROM แบบหนึ่งนั่นเอง) นอกจากนี้พริ้นเตอร์บางแบบยังมีอุปกรณ์เก็บข้อมูลโดยเฉพาะ เช่น ฮาร์ดดิสก์ของพริ้นเตอร์เองต่างหากก็มี

2.3.3 รูปแบบการควบคุม

ข้อมูลที่ส่งจากเครื่องพีซีไปยังพริ้นเตอร์ แต่เดิมจะมีที่อยู่ในรูปรหัสตัวอักษรธรรมดา เช่น รหัส ASCII ซึ่งพริ้นเตอร์จะนำไปเทียบเอารูปฟอนต์ (Font) ที่มีอยู่มาพิมพ์อีกที และที่เป็นรูปกราฟฟิก ข้อมูลที่จะส่งจะแทนแต่ละจุดของรูปที่จะปรากฏบนกระดาษไปเลย ต่อมาลักษณะการพิมพ์จะเพิ่มความซับซ้อนขึ้นโดยมีการพิมพ์ทั้งตัวอักษรและรูปภาพปนกันในแต่ละหน้า ตัวอักษรเองก็เพิ่มความซับซ้อนขึ้น มีฟอนต์ (Font) ต่าง ๆ ให้เลือกใช้มากขึ้น และในแต่ละฟอนต์ (Font) ก็มีขนาดต่าง ๆ กัน มีทั้งตัวหนา (bold) ตัวเอน (italic) และอื่น ๆ อีกมาก ส่วนทางด้านรูปกราฟฟิกก็มีจำนวนจุดเพิ่มมากขึ้นมากขึ้นเพื่อแทนทั้งหลายเส้นและการไล่โทนสีต่าง ๆ ดังนั้นจึงมีการพัฒนารูปแบบการส่งข้อมูลไปพิมพ์ให้มีความเป็นระบบระเบียบมากขึ้น สามารถควบคุมลักษณะการพิมพ์ที่ซับซ้อนขึ้นได้ เกิดเป็นภาษาที่ใช้ในการควบคุมเครื่องพิมพ์ต่าง ๆ กันออกไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาษาที่ใช้ในการควบคุมนี้จะมีแบบง่าย ๆ เช่นสั่งพริ้นเตอร์ให้เปลี่ยนฟอนต์(Font) หรือขนาดตัวอักษรและแบบยาก ๆ ที่สั่งกำหนดการพิมพ์อย่างพิสดาร ซึ่งมาตรฐานที่ใช้กันทั่วไปมีอยู่ 3 กลุ่มคือ

- Escape Sequence ของเครื่องพิมพ์ตระกูล EPSON
- Page Description Language (PDL) เช่นภาษา PostScript ของ Adobe และ printer control Language (PCL) ของ Hewlette-Packard
- Graphical Device Interface(GDI) ที่ใช้ในระบบปฏิบัติการ Microsoft Windows

2.3.4 หน่วยความจำการพิมพ์

พริ้นเตอร์ที่พิมพ์แบบทีละหน้า (page printer) ทั่วไปมักจะต้องการหน่วยความจำ (RAM) จำนวนมากสำหรับเก็บผลลัพธ์สุดท้าย (ภาพที่สร้างขึ้นมา) ก่อนจะเริ่มพิมพ์จริง เช่น 1 MB ในการพิมพ์บนกระดาษขนาด A4 แต่ถ้าเพิ่มเป็นขนาด A3 ก็จะต้องใช้ RAM มากขึ้นกว่าเท่าตัว การพิมพ์ความละเอียดก็มีผลเช่นกัน ในทางปฏิบัติแล้วผู้ผลิตพริ้นเตอร์ก็จะต้องมีวิธีการจัดสรรหน่วยความจำในเครื่อง และเทคนิคต่าง ๆ ในการใช้หน่วยความจำให้น้อยที่สุดในขณะที่ทำงานได้ครบถ้วน

พริ้นเตอร์บางแบบไม่สามารถทำงานได้ตามแบบมาตรฐานแบบใดแบบหนึ่งใน 4 แบบข้างต้น ได้โดยตรงแต่ก็จะต้องมีความสามารถในการทำงานที่เลียนแบบ(Emulate) เป็นพริ้นเตอร์ที่ทำงานตามแบบมาตรฐานดังกล่าวอย่างใดอย่างหนึ่งได้ โดยการใช้คำสั่งที่แผงควบคุมหน้าเครื่องหรืออาจต้องเสียบการ์ดหรือดิสก์เพื่อเพิ่มหน่วยจำและ โปรแกรมเข้าไป

2.3.5 การเชื่อมต่อพริ้นเตอร์

พริ้นเตอร์ส่วนใหญ่ที่ใช้กันมาจะต่อกับพีซีผ่านทางพอร์ตแบบพอร์ตขนาน ซึ่ง เป็นแบบ centronics (Centronics เป็นพริ้นเตอร์ในยุคแรก ๆ ที่ใช้การต่อพอร์ตแบบพอร์ตขนาน และได้กลายมาเป็นชื่อเรียกลักษณะการต่อสัญญาณ) แต่เดิมพอร์ตดังกล่าวมักจะเป็นแบบส่งข้อมูลในทางเดียวคือจากพีซีไปยังพริ้นเตอร์ แต่เนื่องจากในระยะหลังเริ่มมีการผลิตอุปกรณ์เพิ่มเติมสำหรับพีซีที่ใช้ต่อกับพอร์ตขนานมากขึ้น พอร์ตในเครื่องระยะหลังจึงกลายเป็นแบบ 2 ทาง (Bidirectional) คือได้ทั้งรับและส่งข้อมูล การต่อสายแบบขนานนี้สามารถส่งข้อมูลได้ด้วยความเร็วสูงกว่าพอร์ตอนุกรม แต่เนื่องจากใช้แรงดันไฟต่ำกว่า คือใช้เพียง 5 โวลท์ (ในขณะที่แบบอนุกรมใช้ 12 โวลท์) จึงส่งได้ในระยะไกลกว่าคือเพียง 5-6 เมตรเท่านั้น

2.4 ไมโครคอนโทรลเลอร์เบสิกแสดมปี 2พี (2P)

เบสิกแสดมปี (Basic Stamp) เป็นที่รู้จักกันอย่างกว้างขวางมากขึ้นในประเทศไทยในช่วงปี 1999 อันเนื่องมาจากการผลักดันและนำเสนออย่างต่อเนื่องของบริษัทพาราแลกซ์(Parallax Inc.) ผู้ผลิตชิปโมดูลไมโครคอนโทรลเลอร์นี้จากสหรัฐอเมริกาและบริษัทอินโนเวตีฟอิเล็กทรอนิกส์ จำกัดในประเทศไทย เหตุผลที่มีคำว่าแสดมปีต่อท้ายก็เพื่อต้องการให้ทราบว่าบอร์ดไมโครคอนโทรลเลอร์ตัวนี้มีขนาดเล็กเท่ากับแสดมปี

เบสิกแสดมปีเป็นไมโครคอนโทรลเลอร์ที่มีอยู่ด้วยกันหลายเบอร์ ซึ่งแต่ละเบอร์นั้นก็มีคุณสมบัติที่แตกต่างกันตามแต่จะเลือกใช้งานให้เหมาะสม โดยทั่วไปแล้วคุณสมบัติที่แตกต่างในแต่ละเบอร์ ก็คือ จำนวนหน่วยความจำรวม แรม อินพุต เอาต์พุต และโครงสร้างภายในของเบอร์เบสิกแสดมปี 2P

เบสิกแสดมปีเป็นไมโครคอนโทรลเลอร์ที่บรรจุตัวแปลงภาษาเบสิก หรือจะเรียกว่าอีกอย่างหนึ่งคือ เบสิกอินเตอร์พรีเตอร์ (BASIC Interpreter รวมไว้ด้วย สามารถเขียนภาษาเบสิกที่เรียกว่า พีบสิก(PBASIC) ควบคุมการทำงานได้



รูปที่ 2.3 แสดงโครงสร้างเบื้องต้นของเบสิกแสดมปี 2P

2.4.1 คุณสมบัติของไมโครคอนโทรลเลอร์เบสิกแสดมปี 2P

- มีจำนวนขาอินพุตเอาต์พุต 32 ขา (PO-P15 และ A0-A15) และขาสื่อสารอนุกรม 2 ขา (S_{IN} , S_{OUT})
- ต้องการแหล่งจ่ายไฟ 5 โวลต์ ชุดเดียว
- มีจำนวนขาอินพุตเอาต์พุต 40 ขา และขาอนุกรม 3 ขา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ชุดคำสั่งพีเบสิกสอง (PBASIC-2) 55 คำสั่ง สามารถประมวลผลทางคณิตศาสตร์และลอจิกได้
- หน่วยความจำโปรแกรม 16 กิโลไบต์ แบ่งออกเป็น 8 ช่วง ช่วงละ 2 กิโลไบต์ สามารถบรรจุโปรแกรมที่แตกต่างกันได้มากถึง 8 โปรแกรม โดยจะให้ทำงานต่อเนื่องกันหรือแยกกันได้
- สามารถเขียนและลบหน่วยความจำได้ 100,000 รอบ เก็บข้อมูลได้นาน ถึง 10 ปี
- มีหน่วยความจำแรม 160 ไบต์ โดยแบ่งเป็นแรมภายใน 32 ไบต์ และหน่วยความจำสแครตช์แพด (scratch pad RAM) 128 ไบต์
- ความเร็วในการทำงานภาษาพีเบสิกสอง (PSBASIC-2) สูงถึง 12,000 คำสั่งต่อวินาที
- อัตราเร็วในการถ่ายเทข้อมูลกับคอมพิวเตอร์ผ่านพอร์ตอนุกรมสูงถึง 115 กิโลบิตต่อวินาที
- ขาพอร์ตอินพุตเอาต์พุตสามารถใช้ในการกำเนิดสัญญาณพัลส์, สัญญาณ DTMF, สัญญาณ PWM และสามารถใช้ในการวัดความถี่ของสัญญาณไฟฟ้าได้โดยตรง
- ความสามารถในการจ่ายกระแสของขาพอร์ต 30 มิลลิแอมป์ สำหรับกระแสซิงค์และกระแสซอร์สและทุกขาพอร์ตจะสามารถกำหนดให้เป็น ได้ทั้งทางอินพุตและเอาต์พุต
- สามารถขับแอลอีดีและเซอร์โวมอเตอร์ได้โดยตรง
- การ โปรแกรมข้อมูลในลักษณะอนุกรมผ่านทางพอร์ตอนุกรม (RS-232) ของคอมพิวเตอร์
- ความต้องการกระแสไฟฟ้า 65 มิลลิแอมป์ ในขณะที่ทำงานและ 200 ไมโครแอมป์ เมื่ออยู่ในโหมดประหยัดพลังงาน (โหมดสลีป: Sleep mode)

2.4.2 การแบ่งประเภทของหน่วยความจำ

หน่วยความจำหลักที่ใช้กับเบสิกสเตมปี มีอยู่ด้วยกัน 2 ประเภท คือ หน่วยความจำแรมสำหรับเก็บข้อมูลชั่วคราว(RAM Memory) และหน่วยความจำอีพรอมสำหรับเก็บข้อมูลของโปรแกรมพีเบสิกและข้อมูลอื่น ๆ ที่เกี่ยวข้อง (EEPROM Memory)

2.4.2.1 หน่วยความจำแรม

หน่วยความจำสำหรับเก็บข้อมูลชั่วคราวจะหายไปเมื่อไม่มีไฟเลี้ยงวงจรยังสามารถแบ่งได้อีก 2 กลุ่มก็คือ หน่วยความจำสำหรับเก็บค่าตัวแปร(Variable RAM) มีขนาด 32 ไบต์ และหน่วยความจำสแครตช์แพดแรม (Scartchpad RAM) มีขนาด 64 ไบต์

- หน่วยความจำสำหรับเก็บค่าตัวแปร (Variable RAM) อาจเรียกว่า รีจิสเตอร์(Register) เป็นหน่วยความจำหลักที่ไม่โครคอนโทรลเลอร์ต้องมี เนื่องจากในการเขียนโปรแกรมต้องมีการกำหนดค่าตัวแปรเพื่อนำมาทำการประมวลผล และใช้พื้นที่สำหรับเก็บข้อมูลชั่วคราว หน่วยความจำแรมจึงมีบทบาทในส่วนนี้มากในเบสิกแอสเต็มปีสองพี มีหน่วยความจำสำหรับงานลักษณะนี้ 32 ไบต์ โดยจะมีการตั้งชื่อแตกต่างกันเพื่อให้ตัวแปร ภาษาเบสิก ทราบได้ว่า ตัวแปรนี้มีขนาดมากเท่าใดในพื้นที่หน่วยความจำแรม 6 ไบต์แรกได้รับการสงวนไว้สำหรับรีจิสเตอร์พอร์ด เพื่อกำหนดลักษณะการทำงานของชาพอร์ด ทั้ง 40 ขา ส่วนที่เหลืออีก 26 ไบต์ จะเป็นหน่วยความจำอิสระใช้สำหรับเก็บข้อมูลของตัวแปรใด ๆ ในโปรแกรม แต่ต้องพึงระลึกเสมอว่าข้อมูลภายในส่วนนี้ จะสูญหายไปเมื่อไม่มีไฟเลี้ยงวงจร และเมื่อไฟเลี้ยงวงกลับมา ค่าของข้อมูลทั้งหมดจะเป็น 0 หลังจากนั้นจะถูกกำหนดใหม่จากโปรแกรมหลักที่เขียนขึ้น

- หน่วยความจำสแตคซ์แพดแรม (Scartchpad RAM) เป็นพื้นที่พิเศษที่เข้าถึงได้ด้วยการใช้คำสั่ง GET และ PUT ของเบสิกแอสเต็มปีสองพีเท่านั้น โดยพื้นที่ของหน่วยความจำนี้ ใช้สำหรับเก็บข้อมูลชั่วคราวในกรณีที่มีการกระโดดข้ามไปทำงานในโปรแกรมที่อยู่ต่างเพจกันของเบสิกแอสเต็มปีสองพี โดยก่อนการทำงานต้องมีการเก็บข้อมูลของตัวแปรเดิมไว้เสมอด้วยคำสั่ง PUT และเมื่อทำงานในโปรแกรมนั้นเสร็จเรียบร้อยแล้วจะต้องทำการกระโดดกลับมาทำงานที่โปรแกรมเดิมในหน่วยความจำสแตคซ์แพด ก็สามารถเรียกกลับมาได้ด้วยคำสั่ง GET อาจมองได้ว่าหน่วยความจำสแตคซ์แพดคือ สแต็ก (Stack) นั่นเอง โดยมีลักษณะในการเก็บและอ่านข้อมูลแบบเข้าก่อนออกก่อน (FIFO: First-In-First-Out) หรือเก็บก่อนอ่านออกมาก่อน ซึ่งต่างจาก ไมโครคอนโทรลเลอร์ทั่วไปที่จะเป็นแบบ FILO (Firt-In-Last-Out) หรือเก็บเข้าไปก่อนอ่านออกมาทีหลัง โดยใน 63 ไบต์แรกสามารถใช้เก็บข้อมูลได้อย่างเต็มที่ และจะมีการสงวนไว้ 1 ไบต์คือในไบต์สุดท้ายที่ตำแหน่ง 63 สำหรับเก็บหมายเลขโปรแกรมที่เบสิกแอสเต็มปี 2P อินเตอร์พรีเตอร์ทำงานอยู่ในขณะนั้น เพื่อจะได้ทราบว่าขณะนั้นกำลังรันโปรแกรมใดอยู่ ในระหว่างโปรแกรม 0-7 นั้นหมายความว่าหน่วยความจำสแตคซ์แพดที่ตำแหน่งนี้สามารถอ่าน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นับญาติให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ได้เพียงอย่างเดียวเท่านั้นเช่นเดียวกับหน่วยความจำข้อมูลแรม เมื่อเกิดการจ่ายไฟเลี้ยงใหม่หรือเกิดการรีเซ็ต จะทำให้ข้อมูลทั้งหมดในหน่วยความจำกลายเป็น 0 จากนั้นจึงได้รับการกำหนดค่าใหม่จากโปรแกรมที่เขียนขึ้น

2.4.2.2 หน่วยความจำอีอีพรอม (EEPROM Memory)

หน่วยความจำอีอีพรอมนี้จะใช้สำหรับเก็บข้อมูลของโปรแกรมพีเบสิก และข้อมูลอื่น ๆ ที่เกี่ยวข้อง ซึ่งมีความต้องการไม่ให้สูญหายในกรณีที่ไฟเลี้ยงวงจรหายไป เบสิกแอสแตมป์สองพีใช้หน่วยความจำ 16 กิโลไบต์สำหรับเก็บข้อมูลของโปรแกรมพีเบสิกที่แตกต่างกันได้ 8 โปรแกรม โปรแกรมละ 2 กิโลไบต์ ซึ่งต่อไปนี้จะขอเรียกการแบ่งส่วนของหน่วยความจำอีอีพรอมนี้อีกว่า เพจ(Page) ซึ่งก็คือหน่วยความจำอีอีพรอมของเบสิกแอสแตมป์สองพี จะแบ่งออกเป็น 8 เพจ เพจละ 2 กิโลไบต์ อินเทอร์พรีเตอร์ภายในเบสิกแอสแตมป์สองพีจะสามารถเข้าถึงเพื่อรันโปรแกรมได้คราวละ 2 กิโลไบต์เท่านั้น แต่สามารถทำงานทุกโปรแกรมในหน่วยความจำนี้ได้ หากมีการใช้คำสั่งให้กระโดดไปทำงานยังโปรแกรมในตำแหน่งอื่น นั่นคือคำสั่งรัน(RUN) ซึ่งจะได้กล่าวถึงในรายละเอียดต่อไป เมื่อทำการจ่ายไฟ 5 โวลต์ให้ครั้งแรก อินเทอร์พรีเตอร์ภายในเบสิกแอสแตมป์ 2P จะเลือกทำงานที่โปรแกรม 0 ในเพจแรกหรือเพจ 0 เสมอ และสามารถกระโดดไปทำงานยังโปรแกรมอื่นได้ด้วยการบรรจุคำสั่งรัน(RUN) แล้วตามด้วยเลขของโปรแกรมที่ต้องการกระโดดไปทำงาน ซึ่งเมื่อใช้คำสั่งนี้แล้ว จะมีการเก็บตำแหน่งของโปรแกรม เพื่อแจ้งให้โปรแกรมเบสิกแอสแตมป์ 2P อินเทอร์พรีเตอร์ไปทำงานไว้ในหน่วยความจำสแตตซ์แพดแรมในตำแหน่ง 63 ทั้งนี้เพื่อแจ้งให้โปรแกรมเบสิกแอสแตมป์เอ็ดดิเตอร์ ทราบถึงพื้นที่ทำงานของโปรแกรมพีเบสิกขณะนั้น หากมีความต้องการที่จะแก้ไขโปรแกรมตรงเฉพาะในส่วนนั้น ก็สามารถทำได้โดยที่จะไม่จำเป็นต้องดาวน์โหลดหมดทั้ง 16 กิโลไบต์ ของหน่วยความจำอีอีพรอม

TX	1	40	+V
RX	2	39	GND
ANT	3	38	RES
GND	4	37	+5V
P0	5	36	A15
P1	6	35	A14
P2	7	34	A13
P3	8	33	A12
P4	9	32	A11
P5	10	31	A10
P6	11	30	A9
P7	12	29	A8
P8	13	28	A7
P9	14	27	A6
P10	15	26	A5
P11	16	25	A4
P12	17	24	A3
P13	18	23	A2
P14	19	22	A1
P15	20	21	A0

รูปที่ 2.4 แสดงการจัดขาของเบสิกแอสแตมป์ 2P

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.7 แสดงการทำงานในแต่ละขาของเบสิกแอสตมป์2P

ขาที่	ชื่อขา	หน้าที่
1	Tx หรือ SOUT	ส่งข้อมูลออกแบบอนุกรม เชื่อมต่อกับขา R*D (ขา 2) ของพอร์ตอนุกรมของคอมพิวเตอร์ ผ่านคอนเน็กเตอร์ DB-9
2	Rx หรือ S _{IN}	รับข้อมูลเข้าแบบอนุกรม เชื่อมต่อกับขา TD (ขา 3) ของพอร์ตอนุกรมของคอมพิวเตอร์ ผ่านคอนเน็กเตอร์ DB-9
3	ANT	ขาริเซ็คตลจกสูง เชื่อมต่อกับขา DTR (ขา 4) ของพอร์ตอนุกรมของ คอมพิวเตอร์ผ่านคอนเน็กเตอร์ DB-9
4	Vss/GND	ขากราวด์ เชื่อมต่อกับขา GND (ขา 5) ของพอร์ตอนุกรมของคอมพิวเตอร์ ผ่านคอนเน็กเตอร์ DB-9
5 ถึง 20	P0 - P15	ขาพอร์ตอินพุตเอาต์พุต 16 ขา
21 ถึง 36	A0 - A15	ขาพอร์ตอินพุตเอาต์พุตเสริม 16 ขา
37	5 โวลท์	ขาไฟเลี้ยง +5 โวลท์ จากภายนอก
38	RES	ขาริเซ็คตลจกต่ำ สำหรับริเซ็ค I-Stamp2P
39	Vss/GND	ขากราวด์ของ I-Stamp2P
40	Vin	ขาต่อไฟเลี้ยง +6 ถึง +12 โวลท์

2.5 โปรแกรมวิชวลเบสิก 6.0 (Visual Basic 6.0)

โปรแกรม Visual Basic 6.0 เป็นเครื่องมือในการพัฒนาโปรแกรมบน Microsoft Windows 95, Windows 98, Windows NT และ Windows XP ที่ได้รับการพัฒนาโดยบริษัทไมโครซอฟท์ ซึ่งถือเป็นคอมพิวเตอร์ที่ได้รับความนิยมสูง โดยได้เพิ่มความสามารถในการเขียนโปรแกรมติดต่อกับเครือข่ายอินเทอร์เน็ต การเชื่อมต่อกับระบบฐานข้อมูล รวมทั้งปรับปรุงเครื่องมือและการเขียนโปรแกรมเชิงวัตถุ (Object Oriented Programming) ให้สมบูรณ์ยิ่งขึ้นพร้อมทั้งเพิ่มเครื่องมือต่างๆ อีกมากมายที่ทำให้ง่ายและสะดวกขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6 Microsoft Access 2002

Microsoft Access 2002 นั้นเป็นโปรแกรมที่ใช้ในการจัดการฐานข้อมูล ซึ่งจะช่วยให้เราจัดการกับข้อมูลปริมาณมากๆ ได้ง่ายขึ้น ทั้งในการจัดเก็บข้อมูล, การจัดทำรายงานข้อมูล และการสำรองข้อมูลที่เก็บไว้

Access 2002 นั้นเป็นโปรแกรมที่เรียกว่า ระบบจัดการฐานข้อมูล (DBMS : Database Management System) ซึ่งโปรแกรมประเภทนี้ที่เรารู้จักกันก็เช่น dBase, Foxpro, Oracle, SQL Server ซึ่งมีขนาดและขอบเขตการใช้งานที่ต่างกัน Access 2002 นั้นได้รับการพัฒนาจากเวอร์ชันก่อนๆ เพื่อให้เรารองรับงานที่มีข้อมูลจำนวนมากๆ ทั้งงานทั่วไปในสำนักงาน เช่น บัญชี, บุคคล, คลังสินค้า รวมทั้งงานที่เกี่ยวข้องกับอินเทอร์เน็ต นอกจากนี้ Access 2002 ยังมีเครื่องมือต่างๆ ที่ช่วยจัดการฐานข้อมูลอย่างมีประสิทธิภาพ เช่น การใช้ข้อมูลร่วมกับ Access เวอร์ชันเดิม หรือนำข้อมูลจากโปรแกรมฐานข้อมูลชนิดอื่นๆ มาใช้งาน เป็นต้น

2.7 Data Access Object กับการเชื่อมต่อฐานข้อมูล

Data Access Object ต่อไปนี้เรียกว่า DAO เป็นวิธีการเชื่อมต่อระหว่าง Visual Basic กับฐานข้อมูลวิธีหนึ่งจากหลายๆ วิธี DAO ที่นำมาใช้กับฐานข้อมูลนี้ถือว่าเป็นวิธีที่ใช้มานานกับ Visual Basic มีข้อดีคือ ใช้งานง่าย ประสิทธิภาพดีไม่ที่ตัวก็สามารถใช้งานได้เลย แต่มีข้อเสียคือ หากนำไปพัฒนาโปรแกรมในลักษณะเน็ตเวิร์คโดยมีการเชื่อมต่อฐานข้อมูล อาจทำให้โปรแกรมล่าช้า

เนื่องจากการเขียนโปรแกรมโดยใช้ DAO ส่วนใหญ่มักจะใช้คำสั่งภาษา SQL ในการจัดการกับฐานข้อมูล เช่น การดึงข้อมูล การเพิ่มเติม แก้ไข หรือลบข้อมูล เป็นต้น และการเข้าใจภาษา SQL จะช่วยให้สามารถใช้งาน DAO ได้อย่างมีประสิทธิภาพ

บทที่ 3

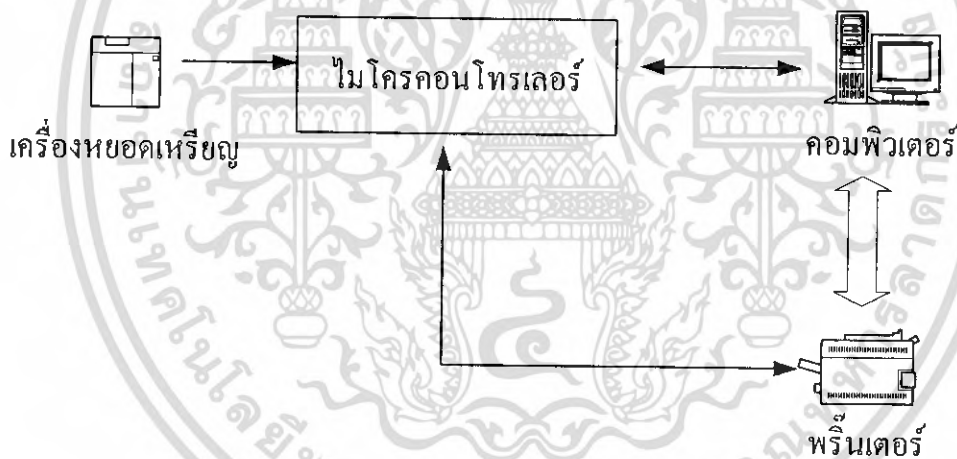
การออกแบบ

การออกแบบเป็นส่วนที่สำคัญเพื่อที่จะได้โปรแกรมที่ทำงานได้ตามขอบเขตที่ต้องการได้ และยังทำงานร่วมกับไมโครคอนโทรลเลอร์ได้ดีที่สุดเราต้องเข้าใจหลักการที่ได้กล่าวถึงในบทที่ 2 และ นำหลักการที่ได้มาออกแบบวงจรที่เราต้องการได้

การออกแบบโครงการจะแบ่งออกเป็น 2 ส่วนคือฮาร์ดแวร์ กับซอฟต์แวร์

1. การออกแบบทางด้านฮาร์ดแวร์(Hardware) ประกอบด้วยส่วนที่ติดต่อกับไมโครคอนโทรลเลอร์ และอุปกรณ์ทางด้านอินพุตเอาต์พุตซึ่งประกอบด้วยเครื่องต่างๆดังนี้เครื่องหอดเหรียญ เครื่องพรีนเตอร์

2. การออกแบบทางด้านซอฟต์แวร์(Software) ประกอบด้วยโปรแกรมที่ใช้สำหรับควบคุมไมโครคอนโทรลเลอร์ เมสิกแสดมปี 2P โปรแกรมที่ใช้สำหรับออกแบบตัวโปรแกรมและติดต่อกับไมโครคอนโทรลเลอร์

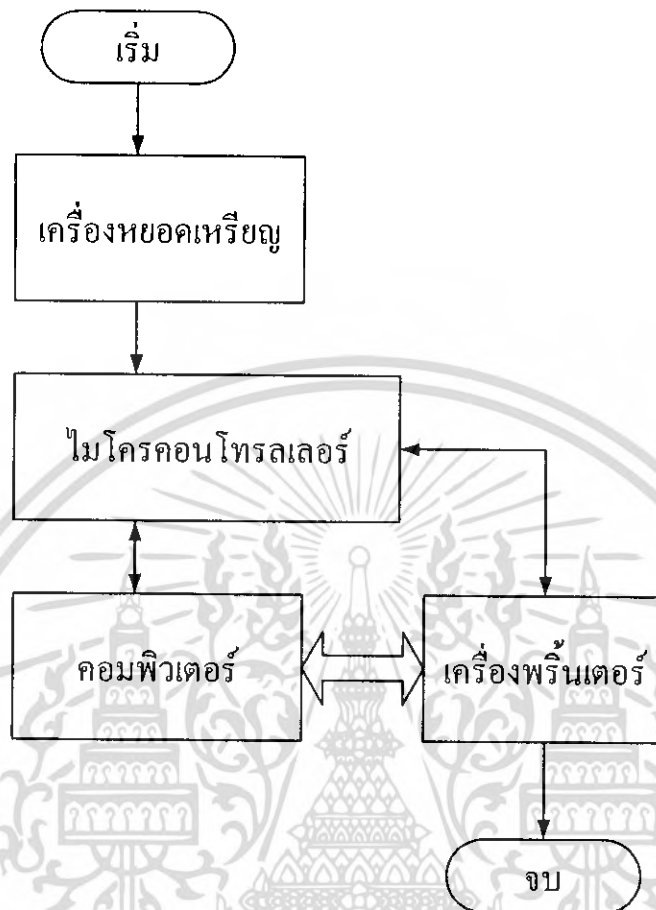


รูปที่ 3.1 เครื่องพรีนเตอร์หอดเหรียญ

การออกแบบเครื่องพรีนเตอร์หอดเหรียญนั้นจะกล่าวถึงการออกแบบโดยรวม การออกแบบแต่ละส่วนเป็นผังงาน(Flowchart) และการออกแบบเพิ่มเติมดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1 การออกแบบโดยรวม



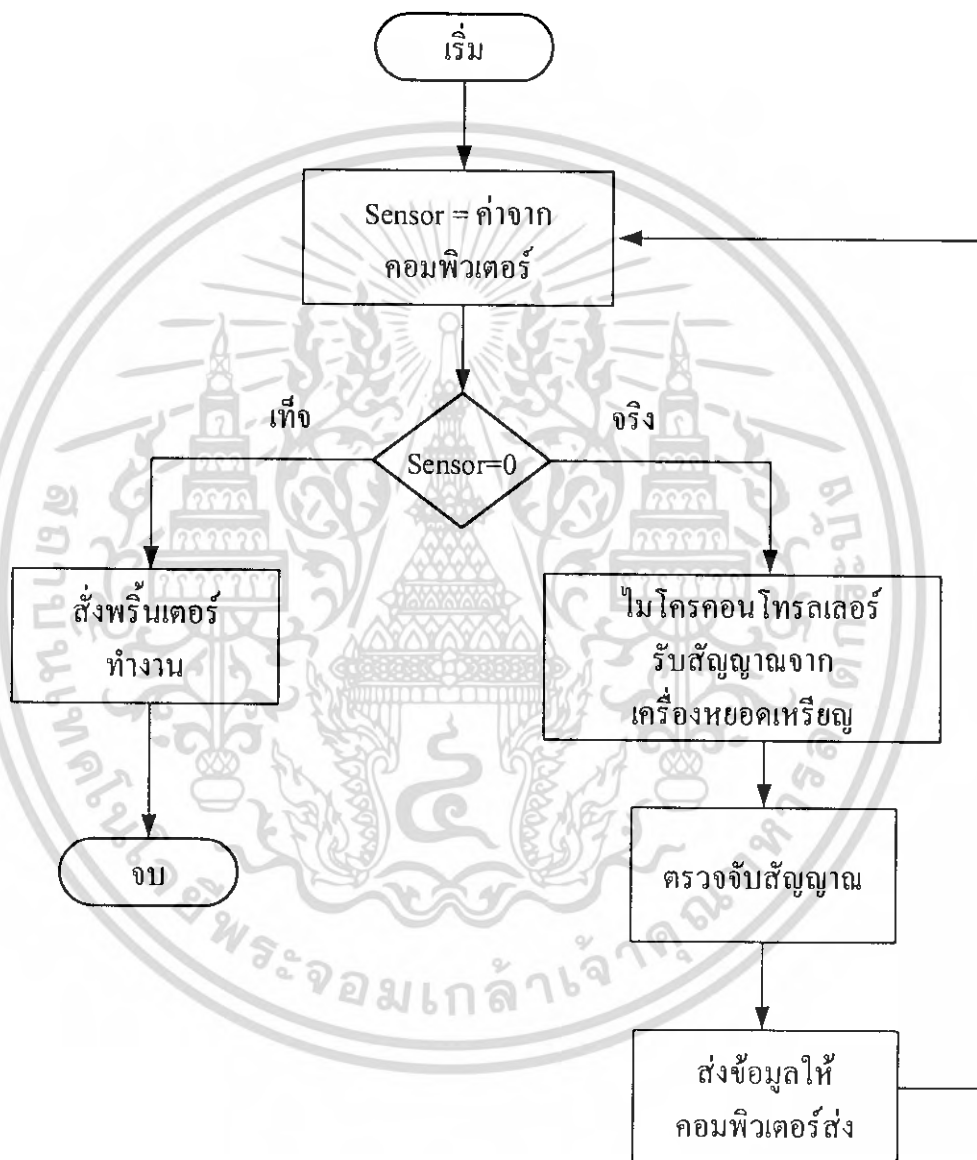
รูปที่ 3.2 การออกแบบโดยรวม

เมื่อเริ่มต้นทำงานเครื่องหยอดเหรียญจะทำการรับเหรียญแล้วตรวจสอบเหรียญจริงหรือปลอม จากนั้นจะส่งสัญญาณพัลส์(จำนวนลูกพัลส์จะเท่ากับราคาของเหรียญ) ไปให้ไมโครคอนโทรลเลอร์ จากนั้นไมโครคอนโทรลเลอร์จะทำการตรวจสอบสัญญาณพัลส์ว่ามีที่ถูกต้องแล้วส่งข้อมูลไปให้คอมพิวเตอร์ คอมพิวเตอร์ได้รับข้อมูลจะทำการคำนวณและแสดงผลหน้าจอมอนิเตอร์เมื่อปุ่มพิมพ์ถูกกดคอมพิวเตอร์ส่งข้อมูลที่ไดจากการคำนวณไปเก็บในตารางข้อมูลแล้วคอมพิวเตอร์จะส่งข้อมูลไปให้กับไมโครคอนโทรลเลอร์อีกครั้งเพื่อไมโครคอนโทรลเลอร์จะส่งสัญญาณให้ทำการเชื่อมต่อพอร์ตขนานของคอมพิวเตอร์กับเครื่องพรีนเตอร์ๆ จะทำการพิมพ์จบการทำงาน

3.2 การออกแบบแต่ละส่วน

การออกแบบการทำงานแต่ละส่วนจะแบ่งออกทั้งหมด 4 ส่วนดังนี้

3.2.1 การออกแบบโปรแกรมควบคุมไมโครคอนโทรลเลอร์ รับสัญญาณจากเครื่องหยุดเหรียญ และส่งข้อมูลให้คอมพิวเตอร์



รูปที่ 3.3 การควบคุมไมโครคอนโทรลเลอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อกำหนดให้ตัวแปร $SW = 0$ ในไมโครคอนโทรลเลอร์ ถ้าคอมพิวเตอรืส่งข้อมูล $SW > 0$ มาไมโครคอนโทรลเลอร์จะสั่งให้เครื่องพรีนเตอร์ทำงานซึ่งจะมีผลให้เครื่องพรีนเตอร์เปิดเครื่องทำงานทำให้สามารถส่งข้อมูลสื่อสารกันได้ แต่ถ้า $SW = 0$ ไมโครคอนโทรลเลอร์จะทำการตรวจสอบจำนวนลูกสัญญาณพัลส์ที่เครื่องหยอดเหรียญส่งมาให้แล้วนับจำนวนพัลส์และส่งจำนวนนั้นไปให้กับคอมพิวเตอรื แล้วรอสัญญาณพัลส์ใหม่

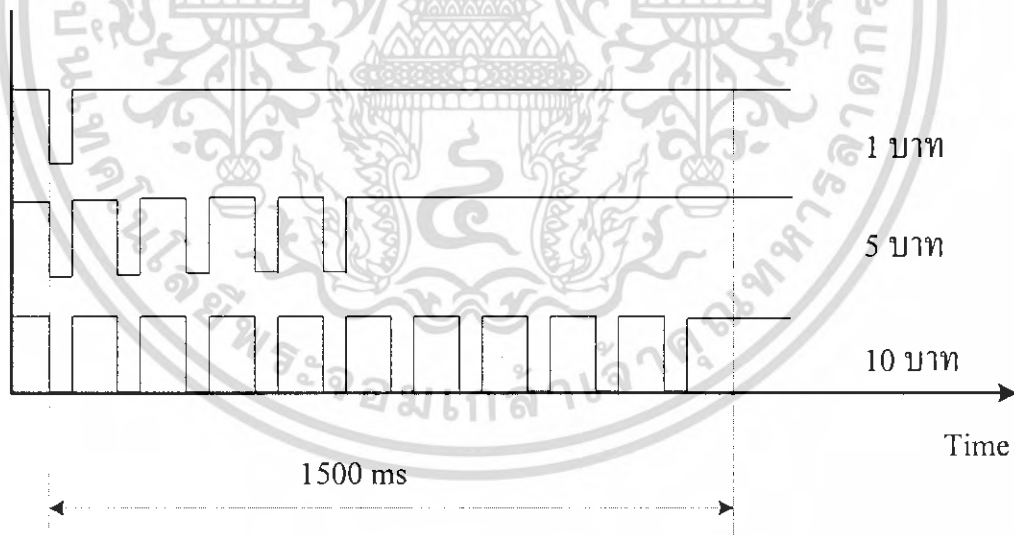
หน้าที่แต่ละส่วนการทำงานของแต่ละส่วนเป็นผังงาน(Flowchart)

สั่งให้เครื่องพรีนเตอร์ทำงาน เมื่อต้องการพิมพ์ สัญญาณจากไมโครคอนโทรลเลอร์จะถูกส่งไปที่เครื่องพรีนเตอร์ให้ทำการเปิดเครื่องพรีนเตอร์ทำงาน



รูปที่ 3.4 สั่งพรีนเตอร์ทำงาน

รับสัญญาณพัลส์จากเครื่องหยอดเหรียญโดยสัญญาณพัลส์จากเครื่องหยอดเหรียญจะมีรูปดังนี้

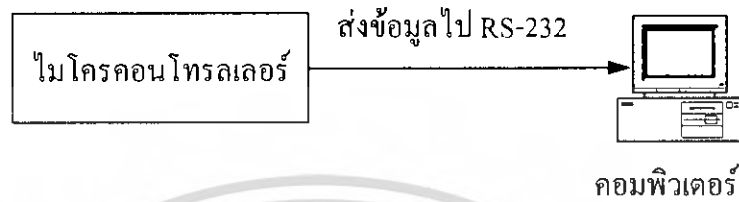


รูปที่ 3.5 สัญญาณพัลส์

ตรวจจับสัญญาณพัลส์จากเครื่องหยอดเหรียญ จะใช้คำสั่ง Count ซึ่งจะนับตามเวลาที่กำหนด รูปแบบการใช้งานคำสั่ง COUNT pin, period, variable (Pin กำหนดขาพอร์ต 0-15 ในที่นี้กำหนดที่ขา 0, Period ช่วงเวลาในการนับ $Period = \text{เวลาที่ต้องการ} / 0.287\text{ms}$, Variable ตัวแปรที่เก็บค่าที่ได้จากการนับ เช่น Count 0, 5227, Cycle) เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่งข้อมูลให้กับคอมพิวเตอร์จะใช้คำสั่ง Serout ซึ่งจะส่งข้อมูลให้กับคอมพิวเตอร์รูปแบบการใช้งานคำสั่ง Serout, tpin, baudmode, [outputdata] (tpin กำหนดขาพอร์ต 0-16 ในการส่งข้อมูลให้คอมพิวเตอร์, baudmode เป็นค่าตัวแปรหรือค่าคงที่ที่จะกำหนดคุณสมบัติต่างๆ ของการสื่อสารอนุกรม, [outputdata] เป็นค่าตัวแปรหรือค่าคงที่ของข้อมูลที่จะใช้ในการส่งเช่น SEROUT 16, 16624, [dec cycles])เป็นต้น



รูปที่ 3.6 ไมโครคอนโทรลเลอร์ส่งข้อมูลให้กับคอมพิวเตอร์

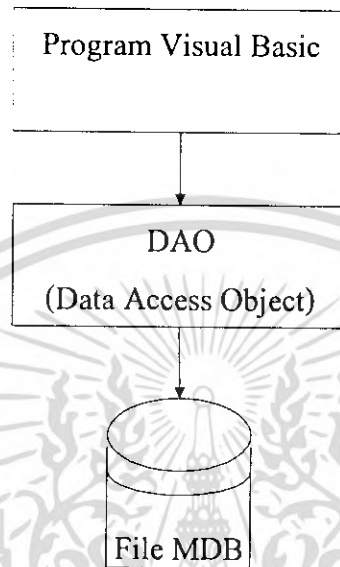
คอมพิวเตอร์ส่งข้อมูลให้กับไมโครคอนโทรลเลอร์จะใช้คำสั่ง Serin ซึ่งจะส่งข้อมูลให้กับคอมพิวเตอร์รูปแบบการใช้งานคำสั่ง Serout, rpin, baudmode, [outputdata] (rpin กำหนดขาพอร์ต 0-16 ใช้ในการส่งข้อมูลให้คอมพิวเตอร์, baudmode เป็นค่าตัวแปรหรือค่าคงที่ที่จะกำหนดคุณสมบัติต่างๆของการสื่อสารอนุกรม, [inputdata] เป็นค่าตัวแปรหรือค่าคงที่ของข้อมูลที่ใช้ในการส่ง เช่นSERIN 16, 16624, [dec cycles])เป็นต้น



รูปที่ 3.7 คอมพิวเตอร์ส่งข้อมูลให้กับไมโครคอนโทรลเลอร์

3.2.2 การออกแบบการติดต่อระหว่าง Visual Basic กับ Microsoft Access


การติดต่อระหว่างโปรแกรม Visual Basic กับฐานข้อมูลจะใช้ DAO มาเป็นตัวที่จะต่อเชื่อมระหว่างโปรแกรมกับฐานข้อมูล โดยที่จะต้องทำการสร้างฐานข้อมูลใน Microsoft Access ก่อน โดยจะเห็นได้ดังรูปข้างล่าง



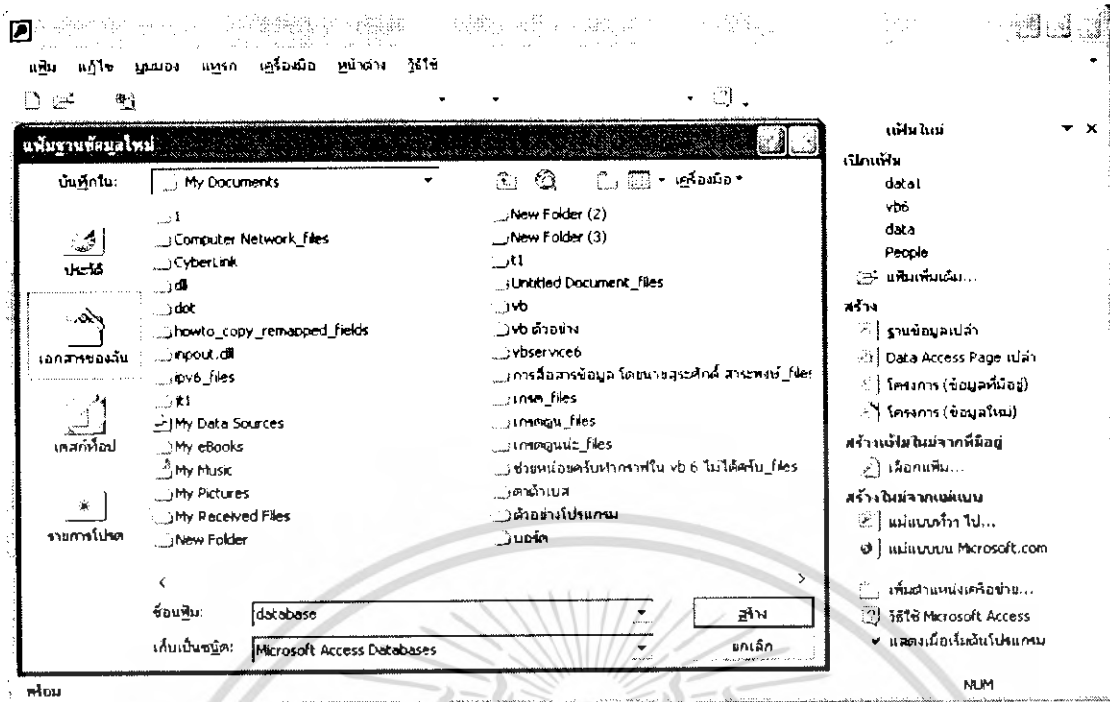
รูปที่ 3.8 การติดต่อระหว่างฐานข้อมูลกับ Visual Basic

ขั้นตอนการสร้างฐานข้อมูลใน Access

ก่อนที่จะสร้างตารางได้เราจะต้องสร้างฐานข้อมูลขึ้นมาก่อน โดยการสร้างฐานข้อมูลด้วยโปรแกรม Access ก็คือ การสร้างไฟล์ของโปรแกรม Access ขึ้นมา ซึ่งไฟล์ที่ได้จะมีนามสกุล *.mdb หลังจากสร้างฐานข้อมูลแล้วเราก็จะได้ฐานข้อมูลเปล่าๆ ที่ยังไม่มีตารางใดๆ อยู่เลย ดังนั้นถ้าต้องการเก็บข้อมูลเราจึงต้องสร้างตารางขึ้นมาเองอีกขั้นตอนหนึ่ง

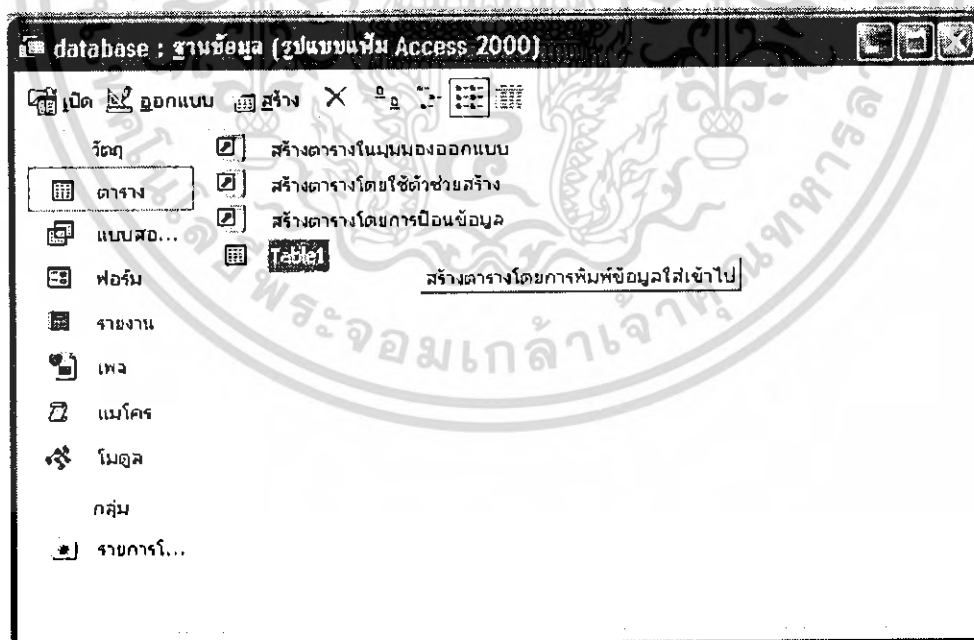
1. เปิดโปรแกรม Access
2. คลิกปุ่ม  ที่ทูลบาร์
3. คลิกคำว่า “ฐานข้อมูลเปล่า” บนหน้าต่าง ซึ่งอยู่ด้านขวาของหน้าต่างโปรแกรม Access
4. เลือกตำแหน่งที่เก็บไฟล์ฐานข้อมูลในเครื่องเรา
5. ตั้งชื่อไฟล์ฐานข้อมูล โดยกำหนดชื่อ database
6. คลิกปุ่มสร้างจากนั้นจะปรากฏหน้าต่างใหม่ขึ้นมา โดยมีชื่อเช่นเดียวกับชื่อฐานข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.9 ขั้นตอนการสร้างฐานข้อมูล

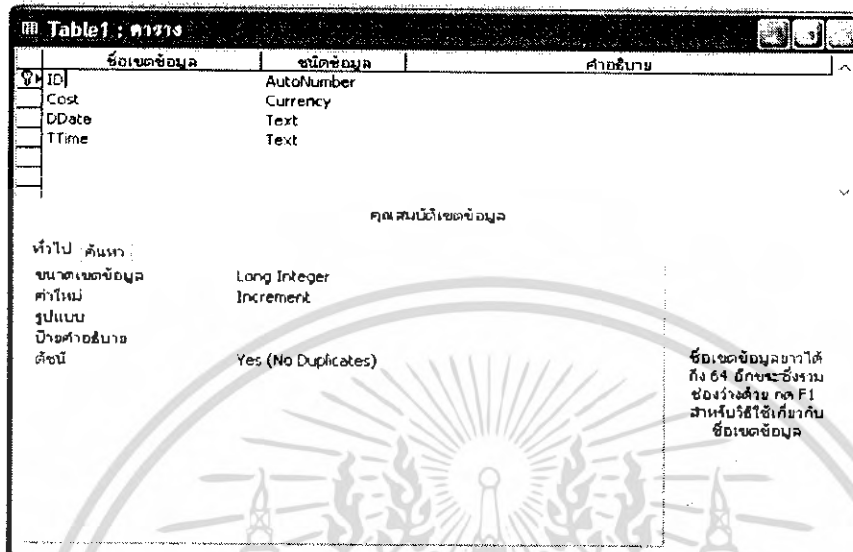
จากนั้นจะได้ฐานข้อมูลเปล่าๆ ที่ยังไม่มีตารางใด โดยเราจะต้องสร้างตารางขึ้นมาที่จะเก็บข้อมูลต่างๆ โดยการสร้างตารางในมุมมองออกแบบขึ้นมาก็จะได้ตาราง ตาราง 1



รูปที่ 3.10 ตารางที่ได้ออกมา

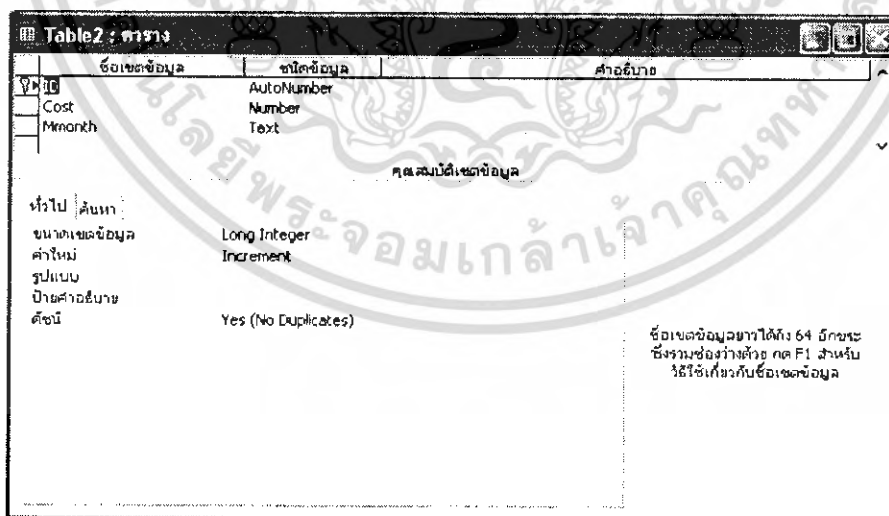
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากนั้นให้เราทำการกำหนดชื่อฟิลด์ข้อมูลในแต่ละคอลัมน์ โดยจะต้องมีชื่อไม่ซ้ำกันและกำหนดชนิดของข้อมูลว่าเป็นชนิดแบบใด และที่สำคัญจะต้องกำหนด Primary key ให้กับ ตารางด้วย โดยจะทำการกำหนดค่า ID เป็น Primary key



รูปที่ 3.11 การกำหนดชื่อฟิลด์ใน ตาราง 1

และทำการสร้างตารางที่ 2 และตารางที่ 3 ในมุมมองการออกแบบอีก โดยกำหนดให้ตารางมีค่า เป็นดังนี้



รูปที่ 3.12 การกำหนดชื่อฟิลด์ใน ตาราง 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชื่อเขตข้อมูล	ชนิดข้อมูล	ค่าเริ่มต้น
Cost	AutoNumber	
Year	Number	
	Text	

คุณสมบัติเขตข้อมูล

ทั่วไป คั่นหน้า:

ขนาดเขตข้อมูล Long Integer

ค่าใหม่ Increment

รูปแบบ

ป้ายค่าอธิบาย

ดัชนี Yes (No Duplicates)

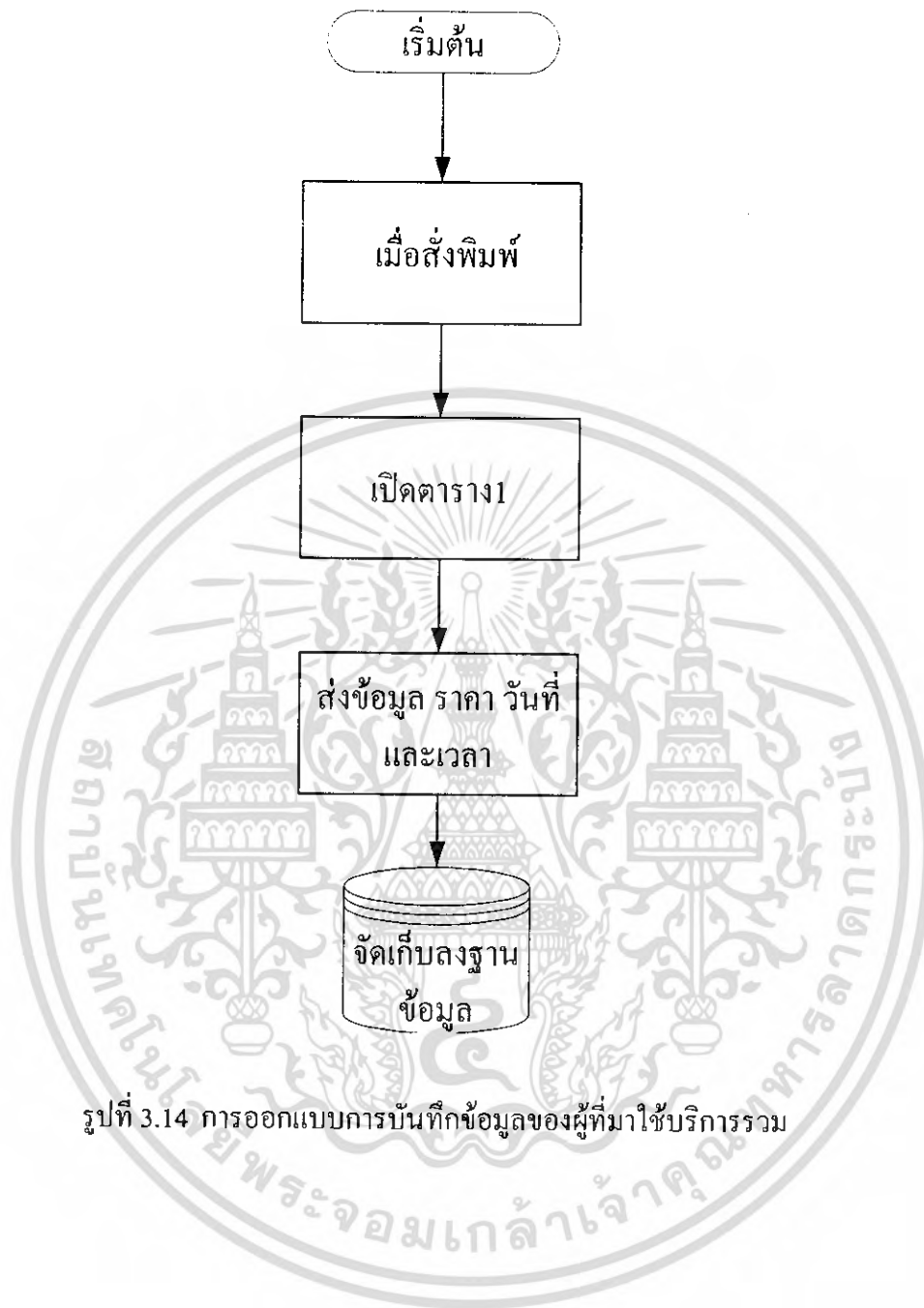
ชื่อเขตข้อมูลยาวได้ถึง 64 อักขระ
ซึ่งรวมช่องว่างด้วย กด F1 สำหรับ
วิธีใช้เกี่ยวกับชื่อเขตข้อมูล

รูปที่ 3.13 การกำหนดชื่อฟิลด์ใน ตาราง 3

เมื่อเราสร้างฐานข้อมูลที่จะจัดเก็บข้อมูลแล้วต่อไปจะเป็นการออกแบบ Flow Chart ในการเขียนโปรแกรมที่ VB โดยทำการติดต่อกับ access

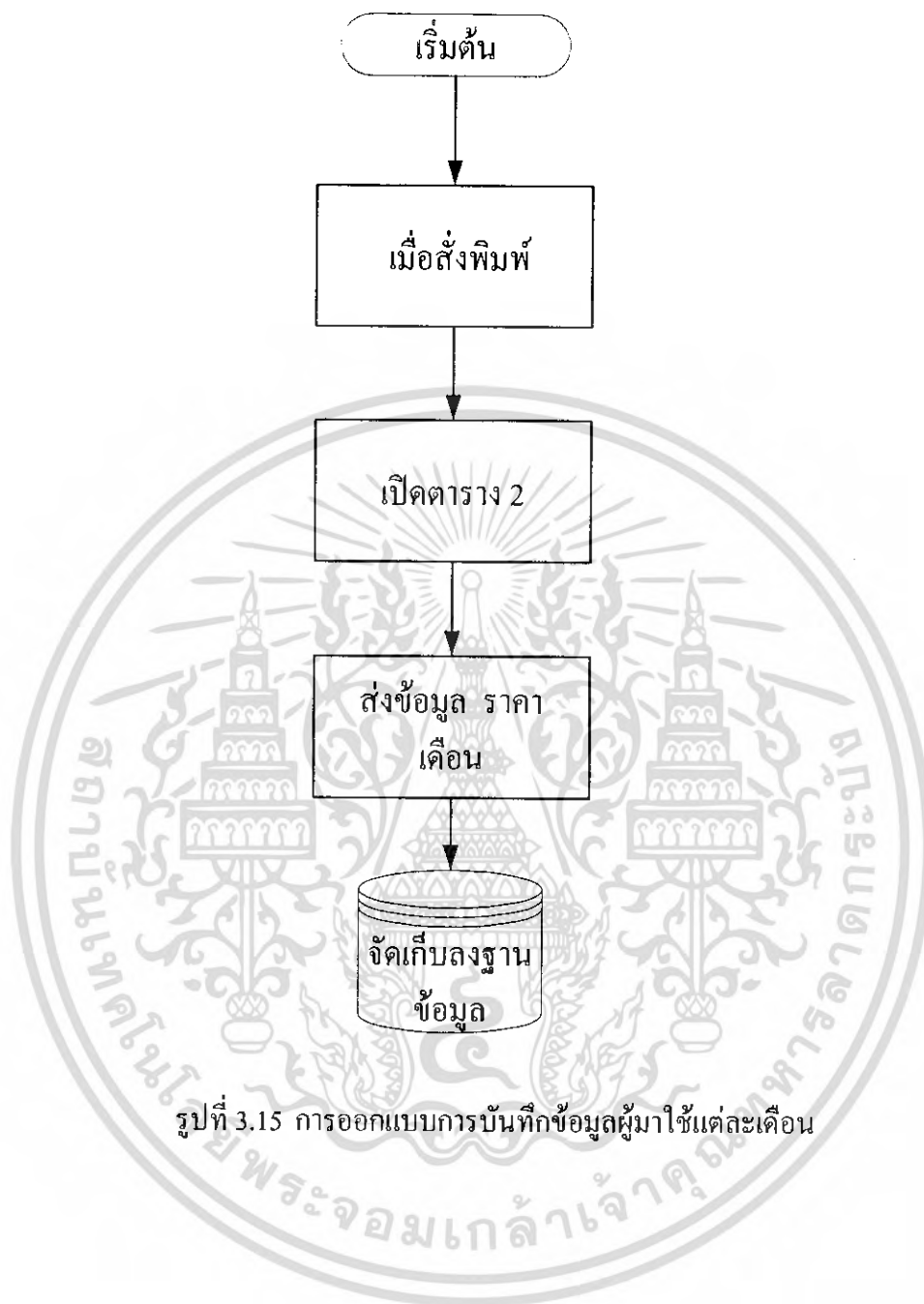
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.2.1 การออกแบบการบันทึกข้อมูลของผู้ที่มาใช้บริการรวม



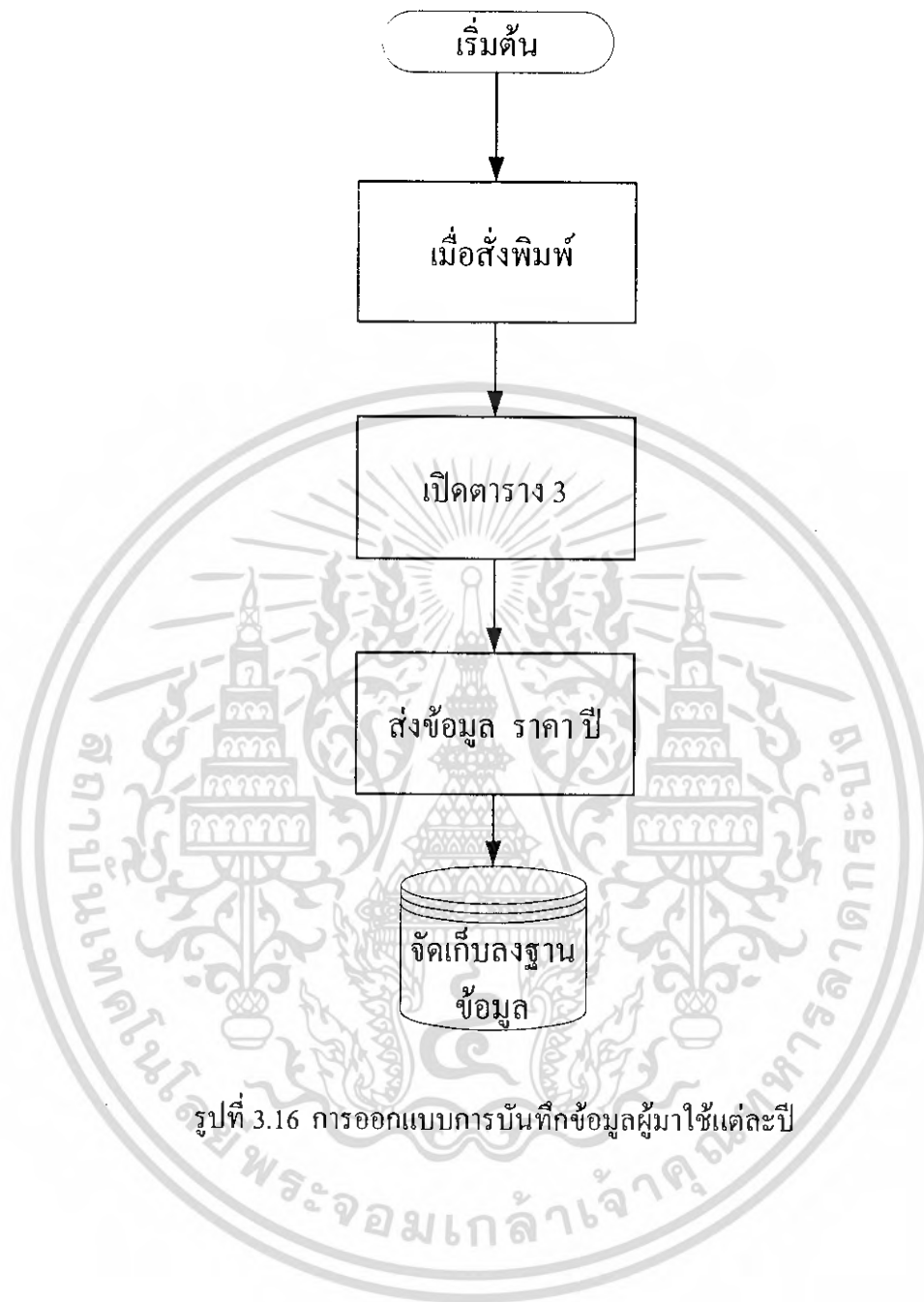
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.2.2 การออกแบบการบันทึกข้อมูลของผู้ที่มาใช้บริการในแต่ละเดือน



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

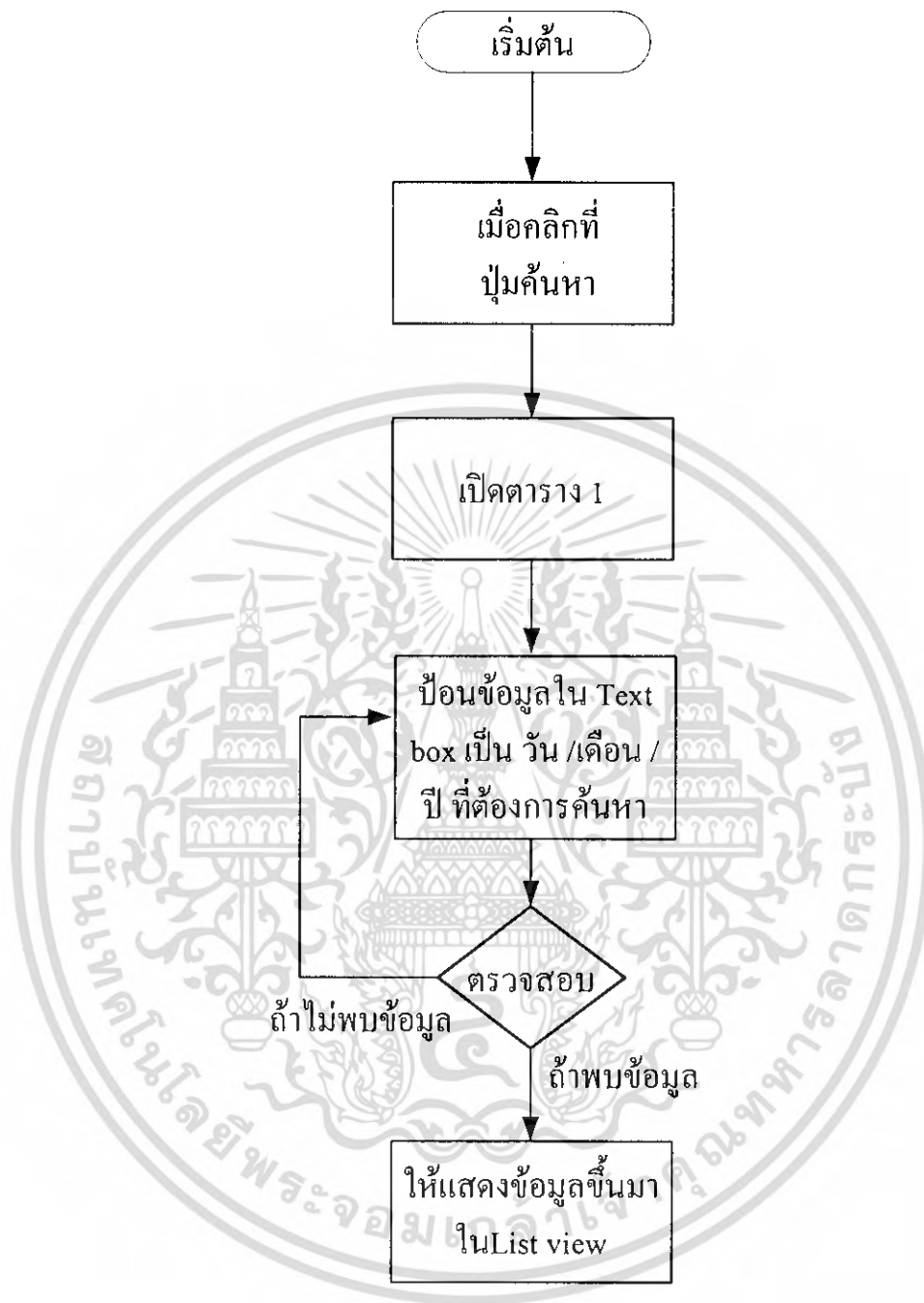
3.2.2.3 การออกแบบการบันทึกข้อมูลของผู้ที่มาใช้บริการในแต่ละปี



รูปที่ 3.16 การออกแบบการบันทึกข้อมูลผู้มาใช้แต่ละปี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

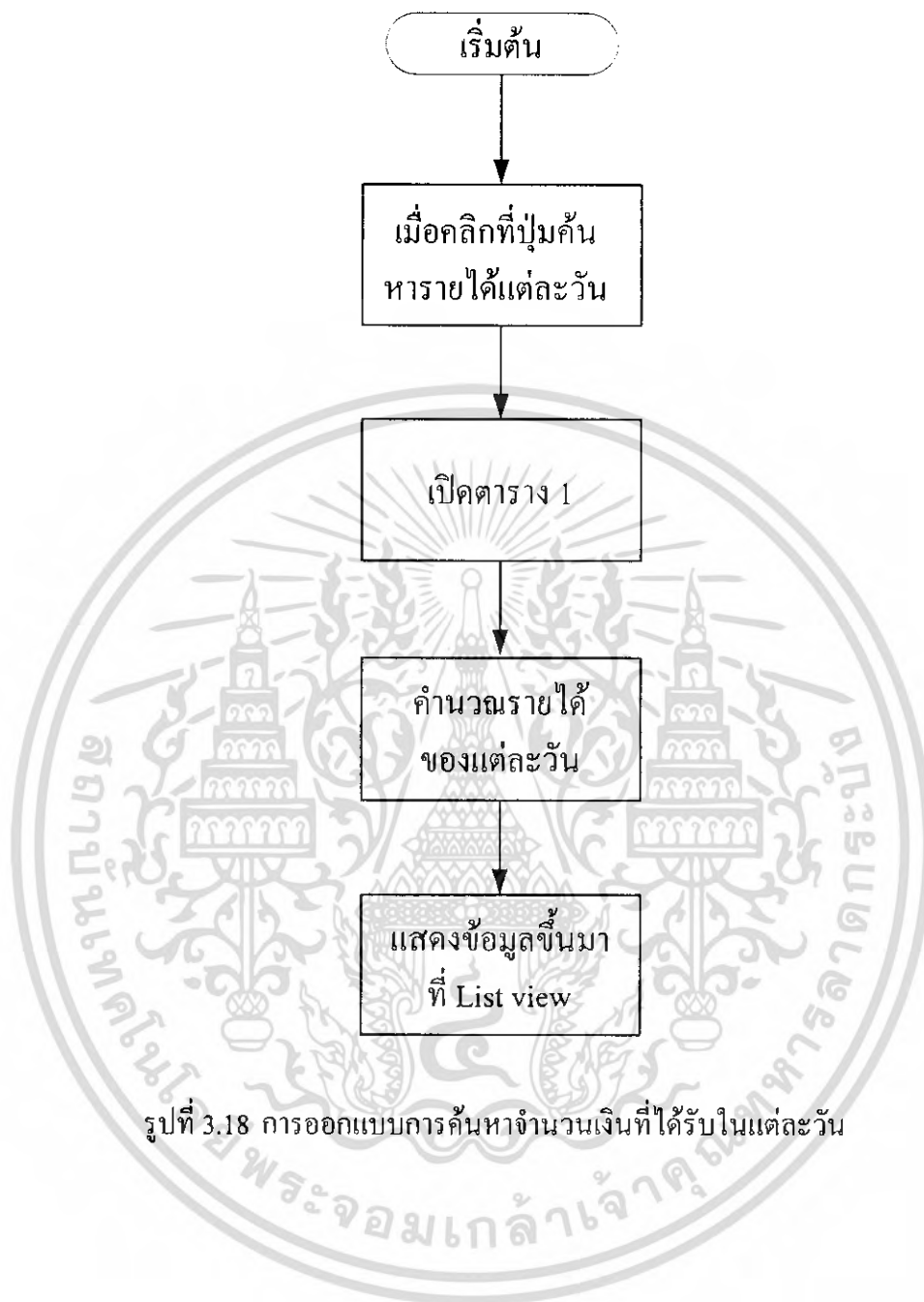
3.2.2.4 การออกแบบการค้นหาข้อมูลผู้มาใช้บริการ



รูปที่ 3.17 การออกแบบการค้นหาข้อมูลผู้ที่ใช้บริการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

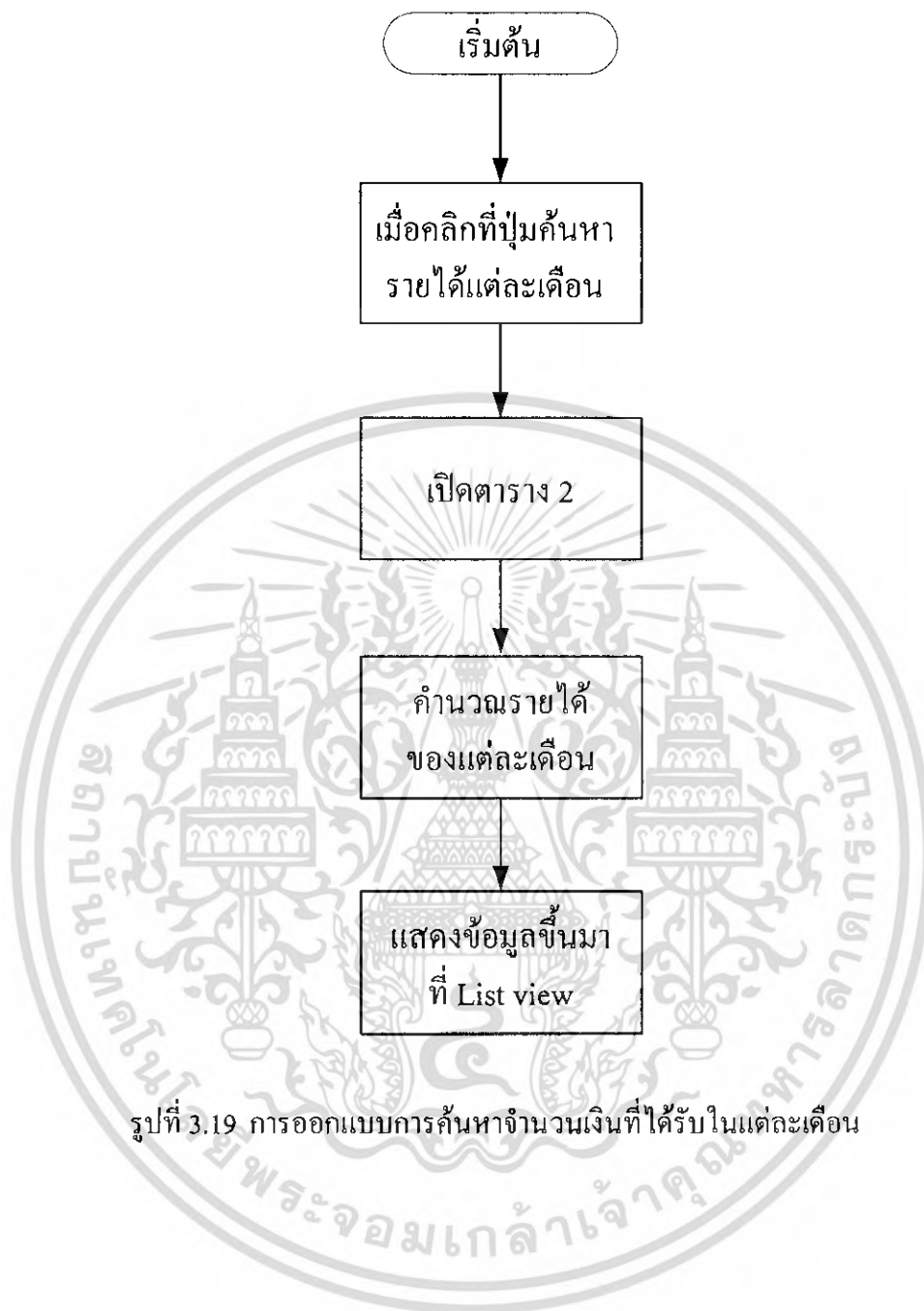
3.2..2.5 การออกแบบการค้นหาจำนวนเงินที่ได้รับในแต่ละวัน



รูปที่ 3.18 การออกแบบการค้นหาจำนวนเงินที่ได้รับในแต่ละวัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

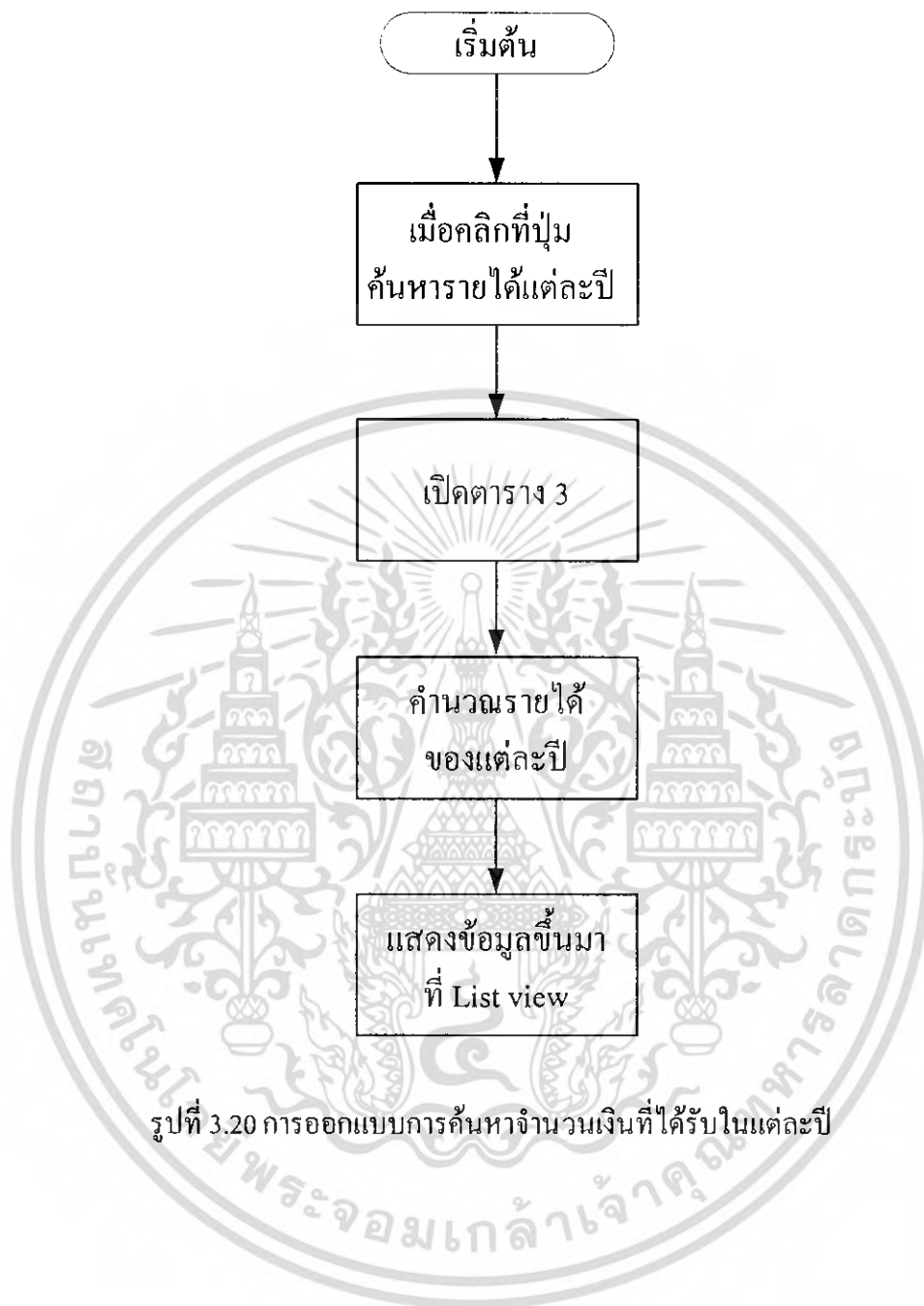
3.2.2.6 การออกแบบการค้นหาจำนวนเงินที่ได้รับในแต่ละเดือน



รูปที่ 3.19 การออกแบบการค้นหาจำนวนเงินที่ได้รับในแต่ละเดือน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.2.7 การออกแบบการค้นหาจำนวนเงินที่ได้รับในแต่ละปี



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อออกแบบ Flow Chart แล้วก็จะทำการเขียนโปรแกรมใน Visual Basic เพื่อทำการติดต่อ

กับ Access

ขั้นตอนการเขียนคำสั่งเพื่อทำงานกับ DAO ใน Visual Basic มีดังนี้

1. ทำการ Reference ไปที่ Microsoft DAO สำหรับ Visual Basic ก่อนที่เราจะประกาศตัวแปรอ็อบเจ็กต์ที่เกี่ยวข้องกับ DAO ก่อนเสมอ

2. ประกาศตัวแปรอ็อบเจ็กต์ต่างๆ สำหรับใช้ร่วมกับ DAO

Dim DB AS Database ตัวแปรสำหรับนำไปเชื่อมต่อกับฐานข้อมูล

Dim RS AS Recordset ตัวแปรสำหรับจะไปใช้กับกลุ่มตารางในฐานข้อมูล

3. สร้างตัวแปรอ็อบเจ็กต์ของ DAO สำหรับเชื่อมต่อกับฐานข้อมูลและตาราง

```
Set DB = DBEngine.Workspaces(0).OpenDatabase(App.Path & "\database.mdb")
```

จากโปรแกรมจะเห็นว่าจะให้เปิดฐานข้อมูล database.mdb

ข้อมูลที่จะไปเก็บในฐานข้อมูล database.mdb จะมี ราคา วันที่ และเวลา โดยมีการเขียน

โค้ดดังนี้

```
MyDate = Format(Now, "DDDDDDDD")
```

กำหนดวันที่ปัจจุบันจากคอมพิวเตอร์

```
MyTime = Format(Now, "Short Time")
```

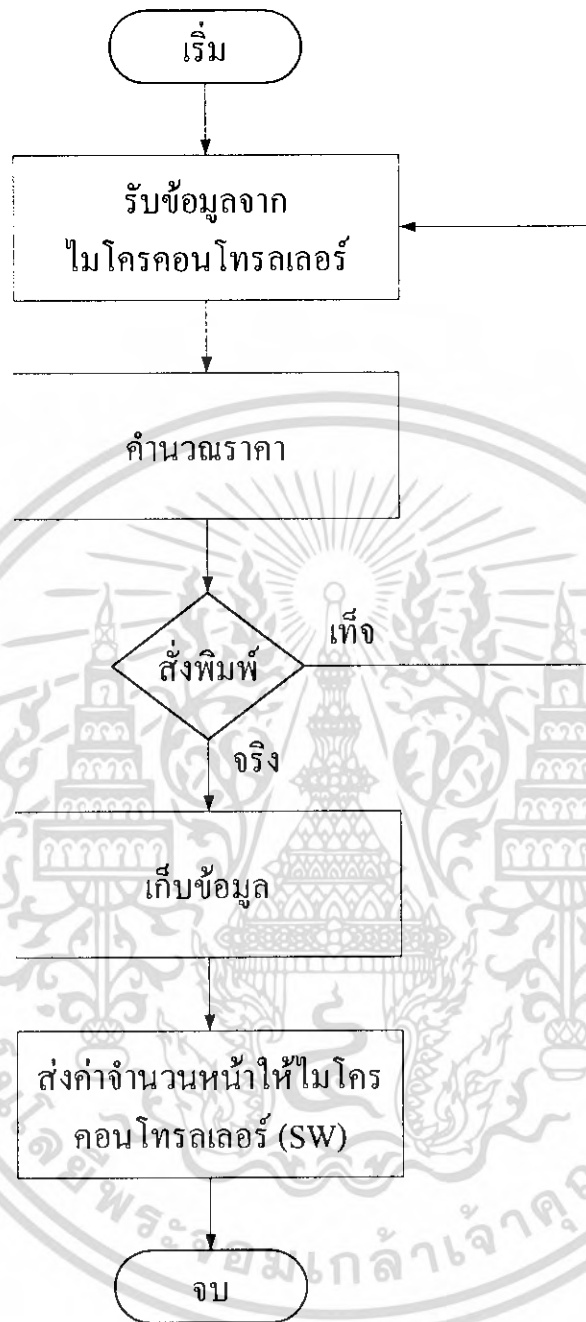
กำหนดเวลาปัจจุบันจากคอมพิวเตอร์

```
sql = "insert into table1 (Cost,DDate,TTime)"
```

```
sql = sql & "values( "" & coin & "" , "" & MyDate & "" , "" & MyTime & "" )"
```

จาก 2 คำสั่งด้านบนคือจะทำการเก็บค่า Cost, MyDate, และ MyTime ไว้ในตัวแปรก่อนที่จะส่งข้อมูลที่เก็บไว้ไปที่ตาราง1 ในแต่ละฟิลด์ของตาราง

3.2.3 การออกแบบโปรแกรมควบคุมการรับ ส่งข้อมูลและสั่งพิมพ์



รูปที่ 3.21 การออกแบบโปรแกรมควบคุมการรับ ส่งข้อมูลและสั่งพิมพ์

เมื่อไมโครคอนโทรลเลอร์ส่งข้อมูลมาให้คอมพิวเตอร์ก็จะนำข้อมูลมาทำการคำนวณว่าขณะนี้เงินที่ขอยกมาเป็นเท่าไรแล้ว ถ้ายังไม่มีคำสั่งพิมพ์จะกลับไปรับข้อมูลที่พอร์ตอนุกรมใหม่ ถ้ามีการสั่งพิมพ์งานจะนำจำนวนเงินไปเก็บไว้ที่ตารางข้อมูล แล้วทำการส่งข้อมูลจำนวนหน้าให้ไมโครคอนโทรลเลอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.4 การนำข้อมูลมาทำกราฟ

เราสามารถนำข้อมูลที่อยู่ใน ตารางข้อมูลมาเสนอในรูปของกราฟ โดยอาศัย Control “Graph” จะใช้คุณสมบัติ GraphType กำหนดรูปแบบของกราฟ

ID	Cost	DDate	TTime
21	฿19.00	27 กันยายน 254	19:44
22	฿17.00	27 กันยายน 254	20:03
23	฿2.00	27 กันยายน 254	20:41
24	17	28 กันยายน 254	13:12
25	฿7.00	28 กันยายน 254	13:33
26	฿10.00	28 กันยายน 254	16:01
27	฿3.00	28 กันยายน 254	16:03
28	฿13.00	28 กันยายน 254	16:08
29	฿3.00	28 กันยายน 254	17:53
30	฿17.00	9 ตุลาคม 25489	15:45
31	฿16.00	9 ตุลาคม 25489	15:48
* (AutoNumber)	฿0.00		

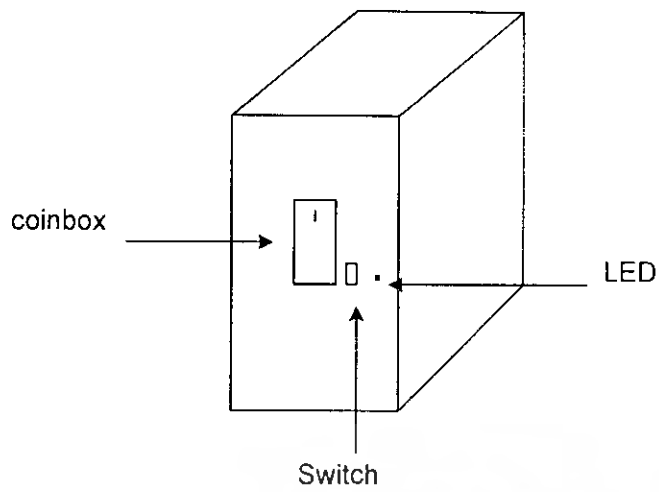
ระเบียน: 14 | 24 | จาก 31

รูปที่ 3.22 บันทึกข้อมูล

3.3 การออกแบบเพิ่มเติม

3.3.1 การออกแบบกล่อง

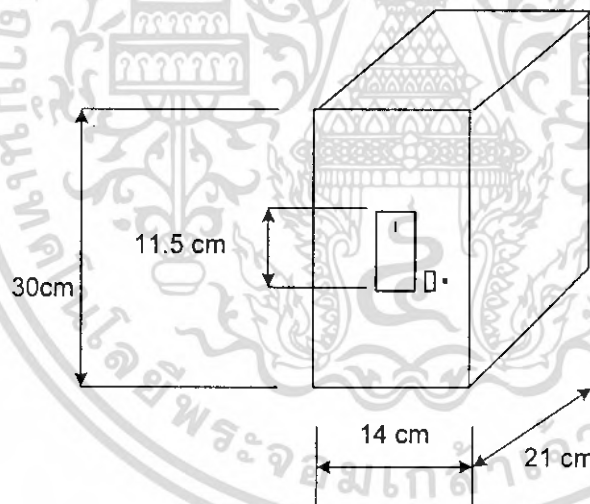
การออกแบบสร้างตู้ที่ใช้ในการเก็บอุปกรณ์ ในการสร้างเครื่องพริ้นเตอร์หยอดเหรียญอัตโนมัติ เครื่องจะต้องสามารถเคลื่อนย้ายได้สะดวกและสามารถเก็บอุปกรณ์ไว้ได้ทั้งหมดเพื่อป้องกันการโดนขโมยเงินหรืออุปกรณ์ต่างๆ ภายในเครื่อง เมื่ออุปกรณ์บางอย่างหายไปอาจจะทำให้เครื่องพริ้นเตอร์หยอดเหรียญตัวนี้ไม่สมบูรณ์และไม่สามารถพิมพ์งานออกมาให้ได้ ฉะนั้นจึงได้มีการออกแบบสร้างตู้ที่ใช้ในการเก็บอุปกรณ์ตู้ขึ้น



รูปที่ 3.23 ตู้เครื่องหยอดเหรียญ

3.3.1.1 การออกแบบด้านหน้าของตู้เครื่องหยอดเหรียญ

การออกแบบด้านหน้าของตู้เครื่องหยอดเหรียญจะต้องเจาะกล่องด้านหน้าสำหรับใส่เครื่องหยอดเหรียญ สวิตช์ และ LED ตามรูป

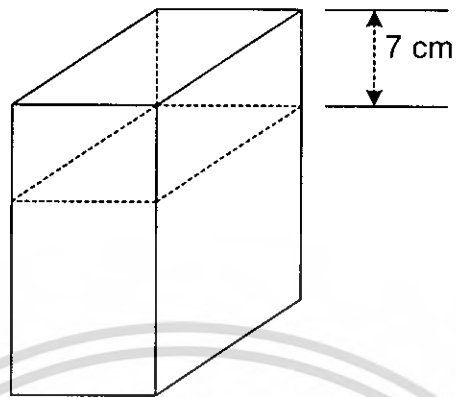


รูปที่ 3.24 ภาพด้านหน้าตู้เครื่องหยอดเหรียญ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.1.2 การออกแบบด้านในของตู้เครื่องหยอดเหรียญ

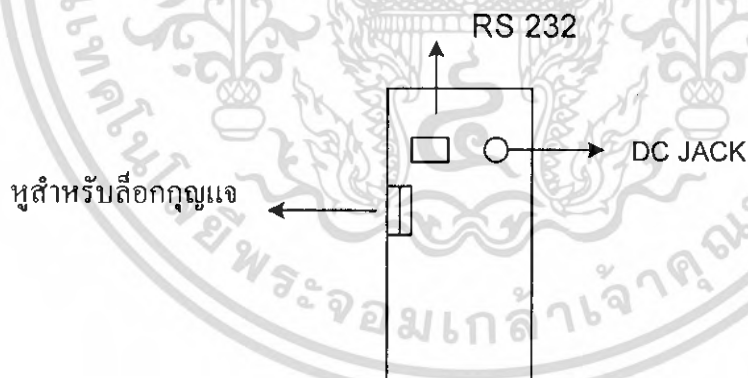
การออกแบบด้านในนั้นจะทำการสร้างชั้นวางวงจรภายในกล่อง ตามรูป



รูปที่ 3.25 ชั้นวางวงจรภายในตู้กล่อง

3.3.1.3 การออกแบบด้านหลังของตู้เครื่องหยอดเหรียญ

การออกแบบด้านหลังของตู้จะทำการเจาะกล่องไว้สำหรับเพื่อใส่พอร์ต RS-232 และ DC JACK และจะมีหูยื่นสำหรับล็อกกุญแจเปิดปิดตู้เครื่องหยอดเหรียญ

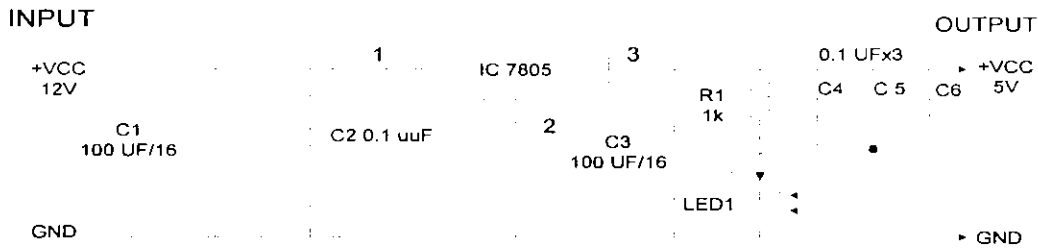


รูปที่ 3.26 การออกแบบด้านหลังตู้

3.3.2 การออกแบบแหล่งจ่ายไฟ 5 โวลต์

การออกแบบแหล่งจ่ายไฟ 5 โวลต์ จะใช้ IC เบอร์ 7805 ในการรีดอปแรงดันไฟอินพุต 12 โวลต์ ให้เหลือ 5 โวลต์ ก่อนที่จะจ่ายให้คอนโทรลเลอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

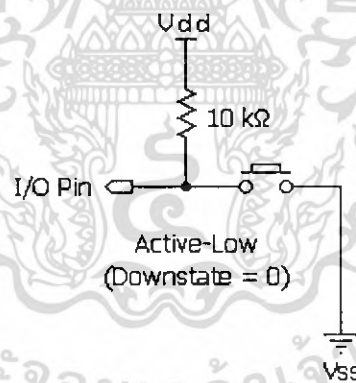


รูปที่ 3.27 ภาคแหล่งจ่ายไฟ 5 โวลท์

3.3.3 การติดต่อเพื่อใช้งานขาพอร์ตของเบสิกแอสตมปี 2P

เบสิกแอสตมปี 2P มีขาพอร์ตให้ใช้งานมากถึง 16 ขาคือ P0-P15 และทุกขาสามารถกำหนดให้ทำงานเป็นขาอินพุตเพื่อรับสัญญาณข้อมูลจากภายนอก หรือเอาต์พุตสำหรับส่งสัญญาณข้อมูลออกไปภายนอกในบทนี้จะกล่าวถึงการใช้งานขาพอร์ตของเบสิกแอสตมปี 2P เป็นขอเอาต์พุตก่อน แต่สิ่งที่จะต้องกล่าวถึงก่อนเป็นอันดับแรกคือ การติดต่อกับขาพอร์ตของเบสิกแอสตมปี 2P

การต่อสัญญาณพัลซ์จากเครื่องหยอดเหรียญเข้าคอนโทรลเลอร์เบสิกแอสตมปี 2P กรณีเลือก ที่ No ที่เครื่องหยอดเหรียญ

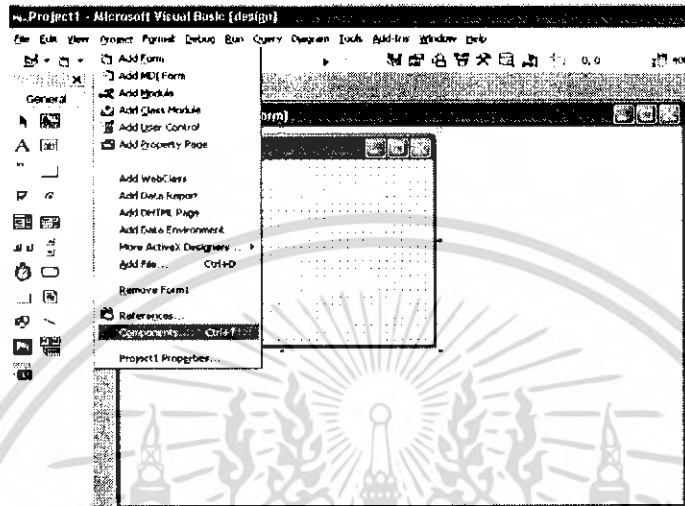


รูปที่ 3.28 การต่อสัญญาณพัลซ์เข้าไมโครคอนโทรลเลอร์เบสิกแอสตมปี 2P

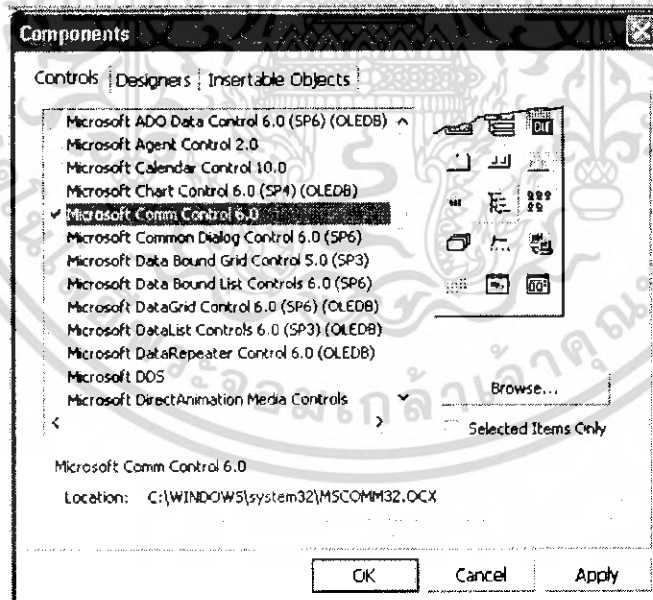
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4 การเขียนโปรแกรมติดต่อ และควบคุมพอร์ตอนุกรม กับ Visual Basic

เนื่องจากในโปรแกรม Visual Basic มีคอนโทรลที่สำคัญคือ MSComm ที่สามารถทำให้ติดต่อ กับพอร์ตอนุกรมได้ เราจึงใช้ตัวคอนโทรลนี้ในการเขียนโปรแกรม



รูปที่ 3.29 ทำการเพิ่มคอนโทรล MSComm

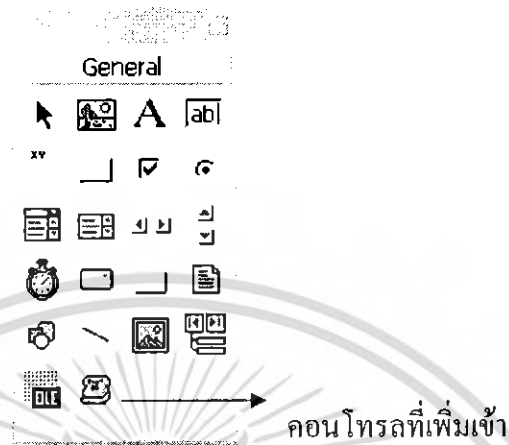


รูปที่ 3.30 เลือกที่รายการ Microsoft Comm Control 6.0

จากนั้นจะปรากฏไอคอนของ Microsoft Comm Control 6.0 ขึ้นมา จากนั้นให้คลิกเลือกที่ Microsoft Comm Control 6.0 แล้วคลิกปุ่ม OK

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากนั้นก็ปรากฏภายใน Toolbox จะมีไอคอนรูปโทรศัพท์ ซึ่งเป็นไอคอนของคอนโทรล MSComm ปรากฏขึ้นมาให้เราเลือกใช้งาน



รูปที่ 3.31 คอนโทรล MSComm พร้อมทำงาน

3.4.1 พร็อพเพอร์ตี้ที่สำคัญในการใช้งาน MSComm

CommPort ใช้ในการกำหนดหมายเลขพอร์ตอนุกรมที่เราต้องการจะติดต่อ โดยมีรูปแบบของการใช้งานดังนี้

Object.CommPort [= value]

ยกตัวอย่างเช่น เรากำหนดให้การเขียนโปรแกรมติดต่อกับพอร์ต Comm1 จะเขียนเป็น

MSComm1.CommPort = 1

Setting ใช้ในการกำหนดอัตราบอด (Baud Rate) หรือความเร็วในการส่งข้อมูลมีหน่วยเป็นบิตต่อวินาที, พาริตี, จำนวนของบิตข้อมูล, จำนวนของบิตปิดท้าย โดยมีรูปแบบของการใช้งานดังนี้

Object.Setting [= value]

ยกตัวอย่างเช่น เรากำหนดให้มีการเขียนโปรแกรมใช้งานที่ Baud Rate = 9600 บิตต่อวินาที ไม่มีพาริตี จำนวนบิตข้อมูลเท่ากับ 8 บิต และมี บิตปิดท้าย 1 บิต

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

PortOpen ใช้สำหรับเปิดและปิดการใช้งานพอร์ตอนุกรม โดยมีรูปแบบของการทำงานดังนี้

```
Objcet.PortOpen [ = value ]
```

ยกตัวอย่างเช่น เราจะเปิดใช้งานพอร์ตอนุกรม ให้กำหนดค่า Value เป็น True เขียนโค้ดได้ดังนี้

```
MSComm1.PortOpen = True
```

แต่ถ้าต้องการปิดพอร์ตอนุกรม ให้กำหนดค่า value เป็น False เช่น

```
MSComm1.PortOpen = False
```

Inputlen เป็นการกำหนดค่าของข้อมูลที่อ่านจาก Buffer ภากรับ โดยมีรูปแบบการกำหนดค่าดังนี้ Object.Inputlen [= value]

Input ใช้ในการอ่านค่าข้อมูลจากพอร์ตอนุกรม โดยมีรูปแบบการอ่านค่าดังนี้

```
Object.Input
```

ยกตัวอย่างเช่น เราอ่านค่าจากบัฟเฟอร์ของพอร์ตอนุกรม แล้วนำมาเก็บไว้ในตัวแปรที่ชื่อว่า Data เราจะเขียนโค้ดดังนี้

```
Data = MSComm1.Input
```

บทที่ 4

ผลการทดลอง

จากการทดลอง

4.1 เปิดงานที่ต้องการพิมพ์ จากโปรแกรมต่างๆ แล้วสำรวจจำนวนหน้าที่ต้องการจะพิมพ์

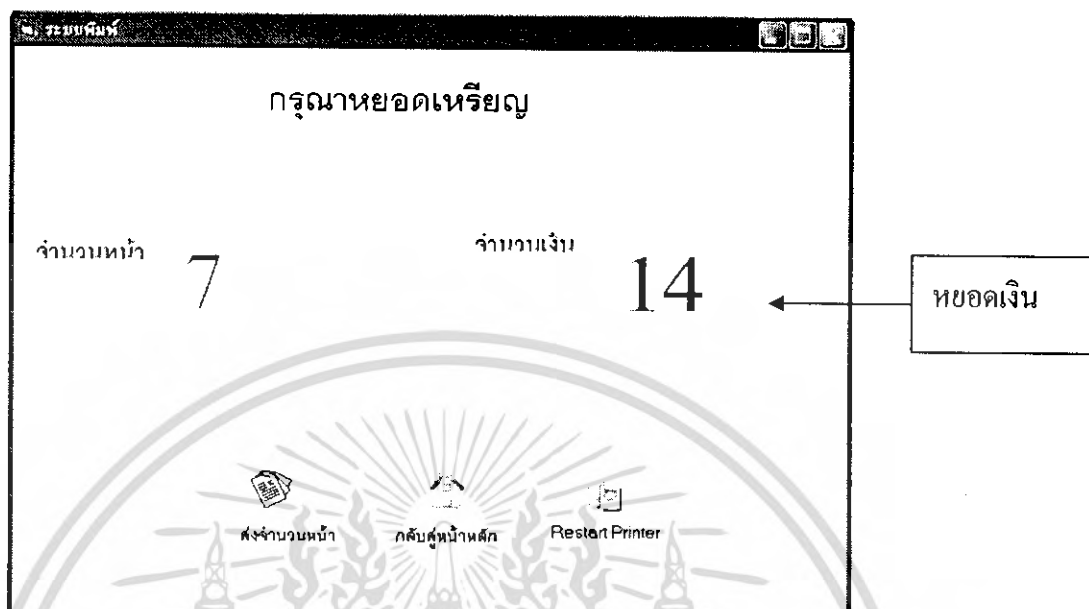


รูปที่ 4.1 สำรวจจำนวนหน้าที่ต้องการจะพิมพ์

4.2 ทำการหยอดเหรียญเมื่อคอนโทรลเลอร์รับค่าจากเครื่องหยอดเหรียญ คอนโทรลเลอร์สามารถเช็คได้ว่าเครื่องหยอดเหรียญรับเหรียญ 1 บาท 5 บาท และ 10 บาท ซึ่งได้ผลการทดลองที่ถูกต้องและเป็นจริงก่อนที่จะส่งเข้าสู่คอมพิวเตอร์โดยให้โปรแกรม Visual Basic

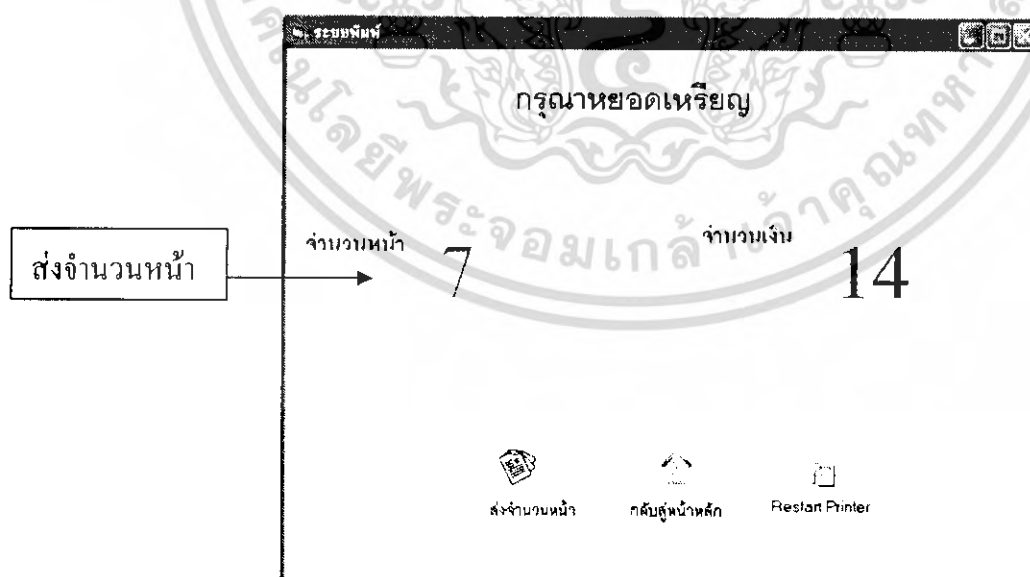
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการแสดงผลว่ารับเหรียญใดเข้ามา ซึ่งในการทดลองนี้ได้ทำการหยอดเงินเป็นจำนวน 14 บาท ค่าที่แสดงออกมาเป็นดังรูป



รูปที่ 4.2 แสดงการหยอดเงิน

4.3 จากนั้นส่งจำนวนหน้าที่โปรแกรมหยอดเหรียญ คอนโทรลเลอร์จะทำการรับค่าจำนวนหน้ากระดาษจากโปรแกรมหยอดเหรียญมาเก็บไว้พร้อมกับสั่งให้เครื่องพรีนเตอร์ทำงาน



รูปที่ 4.3 ส่งจำนวนหน้าที่ที่โปรแกรมหยอดเหรียญ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.5 เมื่อเราทำการส่งพิมพ์ก็จะทำการส่งค่าจำนวนเงิน, วันที่ และเวลาที่นำไปเก็บลงในฐานข้อมูลที่เราได้ทำการสร้างไว้

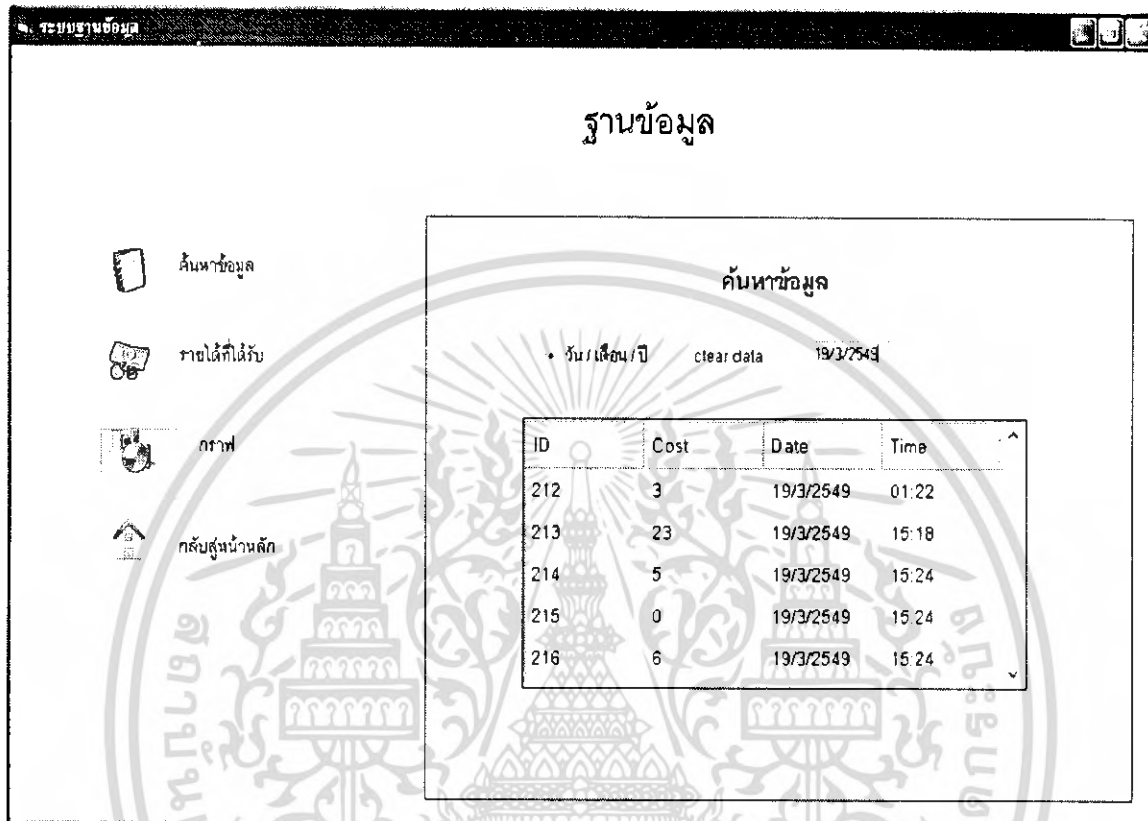
ID	Cost	DDate	TTime
21	฿19.00	27 กันยายน 254	19:44
22	฿17.00	27 กันยายน 254	20:03
23	฿2.00	27 กันยายน 254	20:41
24	17	28 กันยายน 254	13:12
25	฿7.00	28 กันยายน 254	13:33
26	฿10.00	28 กันยายน 254	16:01
27	฿3.00	28 กันยายน 254	16:03
28	฿13.00	28 กันยายน 254	16:08
29	฿3.00	28 กันยายน 254	17:53
30	฿17.00	9 ตุลาคม 25489	15:45
31	฿16.00	9 ตุลาคม 25489	15:48
* (AutoNumber)	฿0.00		

ระเบียน: 14 | 24 | จาก 31

รูปที่ 4.5 นำข้อมูลไปเก็บใน Access

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

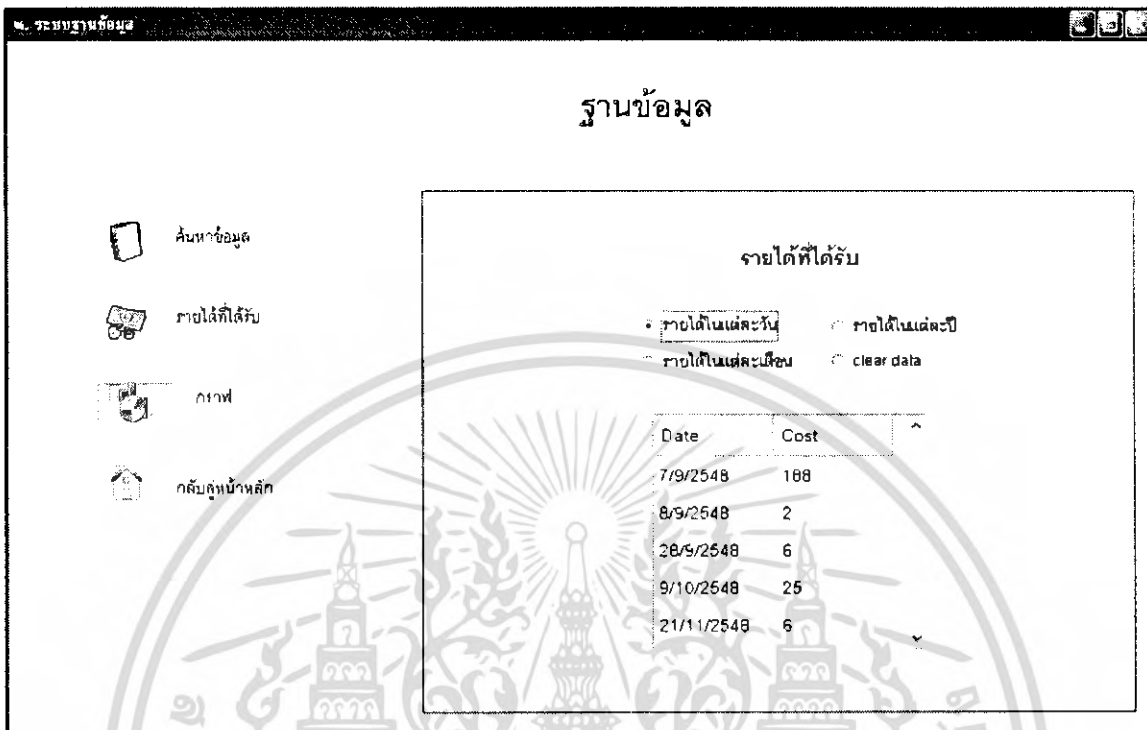
4.6 การค้นหาข้อมูลจะทำการค้นหาข้อมูลที่ถูกจัดเก็บไว้ในฐานข้อมูล เมื่อเรากดคลิกที่ค้นหาข้อมูล ก็จะปรากฏฟอร์มการค้นหาข้อมูลขึ้นมาจากนั้นให้ทำการคลิกที่ วัน/เดือน/ปี โดยจะสามารถค้นหาได้จากการพิมพ์ข้อความเข้าไปที่ text box โดยพิมพ์ วัน/เดือน/ปี เข้าไป



รูปที่ 4.6 การค้นหาข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

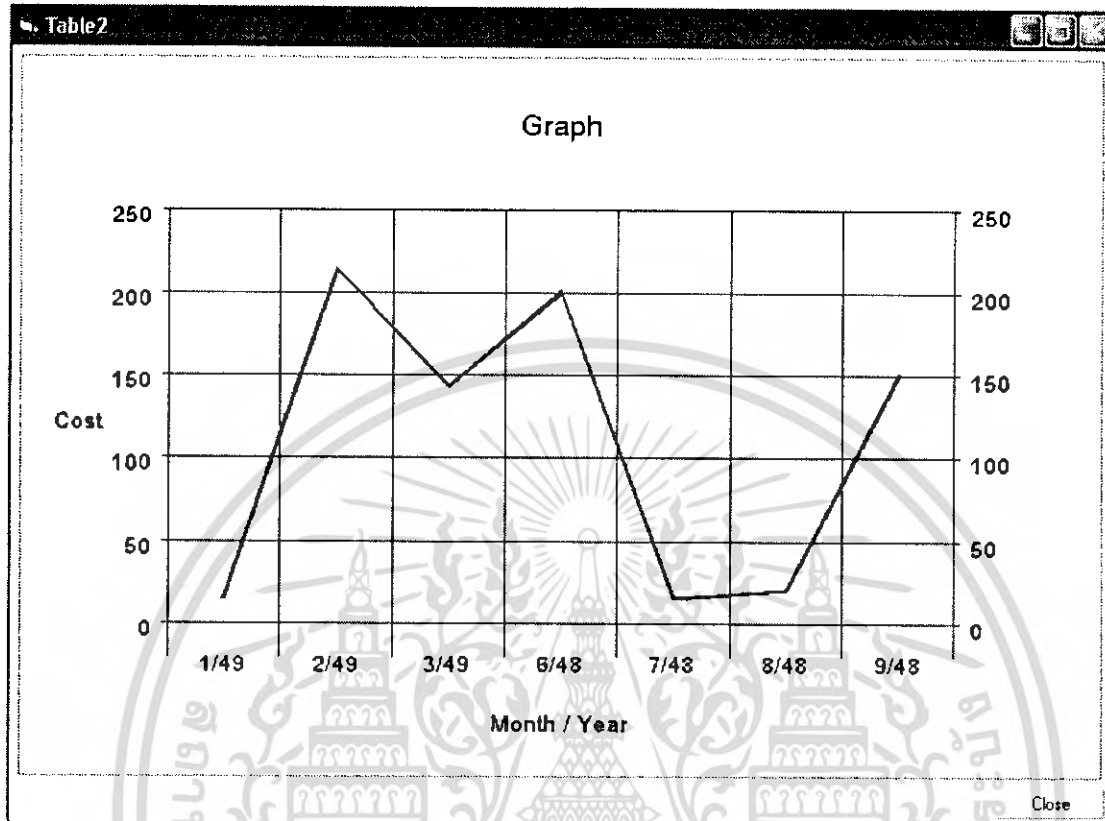
4.7 การค้นหารายได้ที่ได้รับ จะต้องคลิกที่รายได้ที่ได้รับจากนั้นหน้าจอจะปรากฏให้เลือกรายได้ที่ได้รับในแต่ละวัน แต่ละเดือน และแต่ละปี



รูปที่ 4.7 การค้นหารายได้ที่ได้รับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.8 การแสดงกราฟ เมื่อเรากดคลิกที่ปุ่มกราฟ จะแสดงกราฟรายได้ที่ได้รับของแต่ละเดือนในแต่ละปี



รูปที่ 4.8 กราฟรายได้ของแต่ละเดือน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

สรุปผลของโครงการ

5.1 บทนำ

จากที่ได้ค้นทำการออกแบบ สร้างเครื่องพรีนเตอร์หยอดเหรียญ และได้ทำการทดลองโครงการนี้แล้ว ในบทนี้ก็จะเป็นการกล่าวสรุปตลอดจนบอกถึงปัญหาและอุปสรรคต่างๆในการทำโครงการ รวมทั้งข้อเสนอแนะในการพัฒนาโครงการเครื่องพรีนเตอร์หยอดเหรียญต่อไปในอนาคต เพื่อเป็นแนวทางสำหรับผู้ที่เกี่ยวข้องและต้องการที่พัฒนาโครงการนี้ต่อไป

5.2 สรุปผลของโครงการ

โครงการเครื่องพรีนเตอร์หยอดเหรียญจะแบ่งออกเป็น 2 ส่วนคือ ส่วนของฮาร์ดแวร์ ไมโครคอนโทรลเลอร์เบสิกแอสเอ็มบี 2p สามารถตรวจจับสัญญาณจากเครื่องหยอดเหรียญ สามารถส่งค่าข้อมูลให้คอมพิวเตอร์เพื่อใช้ในการคำนวณ และยังสามารถสั่งงานให้เครื่องพรีนเตอร์ทำงานได้เป็นอย่างดี ส่วนของซอฟต์แวร์ ก็จะประกอบด้วยโปรแกรมที่ใช้เก็บข้อมูล โปรแกรมที่ใช้รับควบคุมไมโครคอนโทรลเลอร์เบสิกแอสเอ็มบีสองพี 2p สามารถที่จะเก็บรวบรวมข้อมูลที่ต้องการไว้ในเครื่องคอมพิวเตอร์และยังสามารถเรียกดูข้อมูลที่ต้องการได้เป็นอย่างดี

5.3 แนวทางการพัฒนาต่อ

1. สามารถใช้งานเครื่องพรีนเตอร์หยอดเหรียญได้ง่ายขึ้น
2. สามารถใช้กับเครื่องพรีนเตอร์ได้ทุกเครื่อง
3. สามารถทอนเหรียญได้

บรรณานุกรม

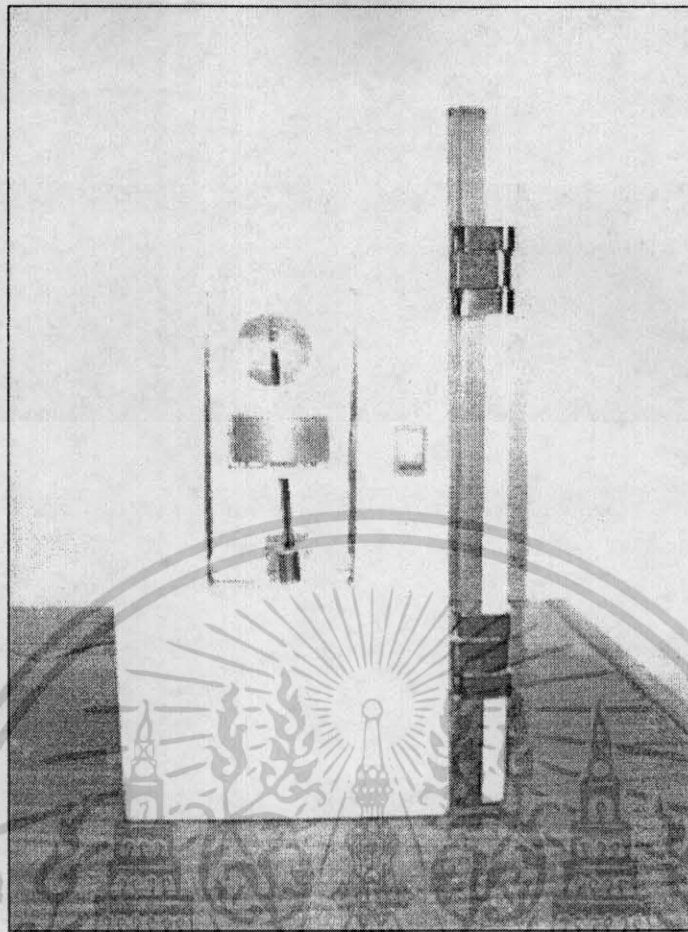
- [1] ณีภูชวพล วงศ์วรชัย, วรพล กรแก้วและชัยวัฒน์ ลี้มพรจิตรวิไล. “เรียนรู้และปฏิบัติการไมโครคอนโทรลเลอร์ด้วยภาษาเบสิกกับเบสิกแสตมป์สองพี(BasicStam2P)”. บริษัทอินโนเวทีฟ เอ็กเพอริเมนต์ จำกัด. พิมพ์ครั้งที่1. 2546
- [2] วรชัย กรแก้ว และชัยวัฒน์ ลี้มพรจิตรวิไล. “คุณสมบัติทางฮาร์ดแวร์และชุดคำสั่งของเบสิกแสตมป์สองเอสเอ็กซ์(BasicStamp 2SX)”. บริษัทอินโนเวทีฟ เอ็กเพอริเมนต์ จำกัด. พิมพ์ครั้งที่ . 2545
- [3] กิตติ ภัคดีวัณณะกุล และจำลอง ครุอุตสาหะ. “วิชวลเบสิก 6 ฉบับโปรแกรมเมอร์”. บริษัท เอ-บู๊คดีสทริบิวชั่น จำกัด. พิมพ์ครั้งที่ 11. 2546
- [4] ศุภชัย สมพานิช. “วิชวลเบสิกโปรแกรมมิ่งฉบับมืออาชีพ”. บริษัทไอทีซี อินโฟคิสทริบิวเตอร์ เซ็นเตอร์ จำกัด. พิมพ์ครั้งที่ 1. 2547

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก

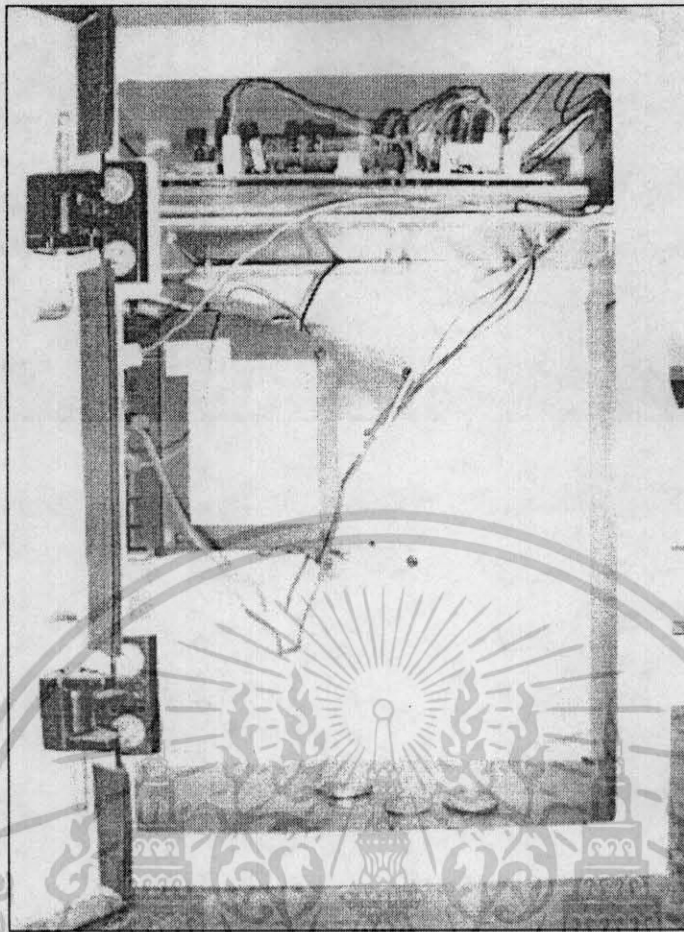


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



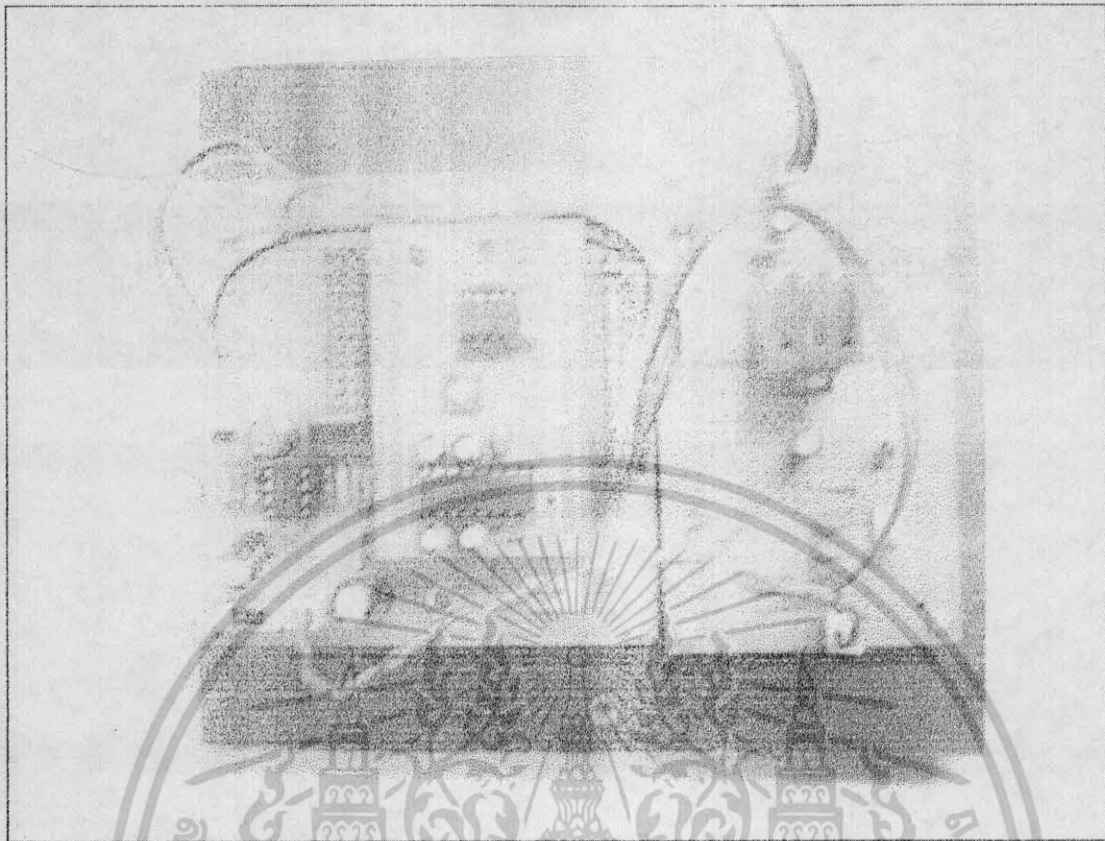
รูปที่ ก.1 ภาพด้านหน้าเครื่องหยอดเหรียญ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ก.2 ภาพด้านในเครื่องหยอดเหรียญ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

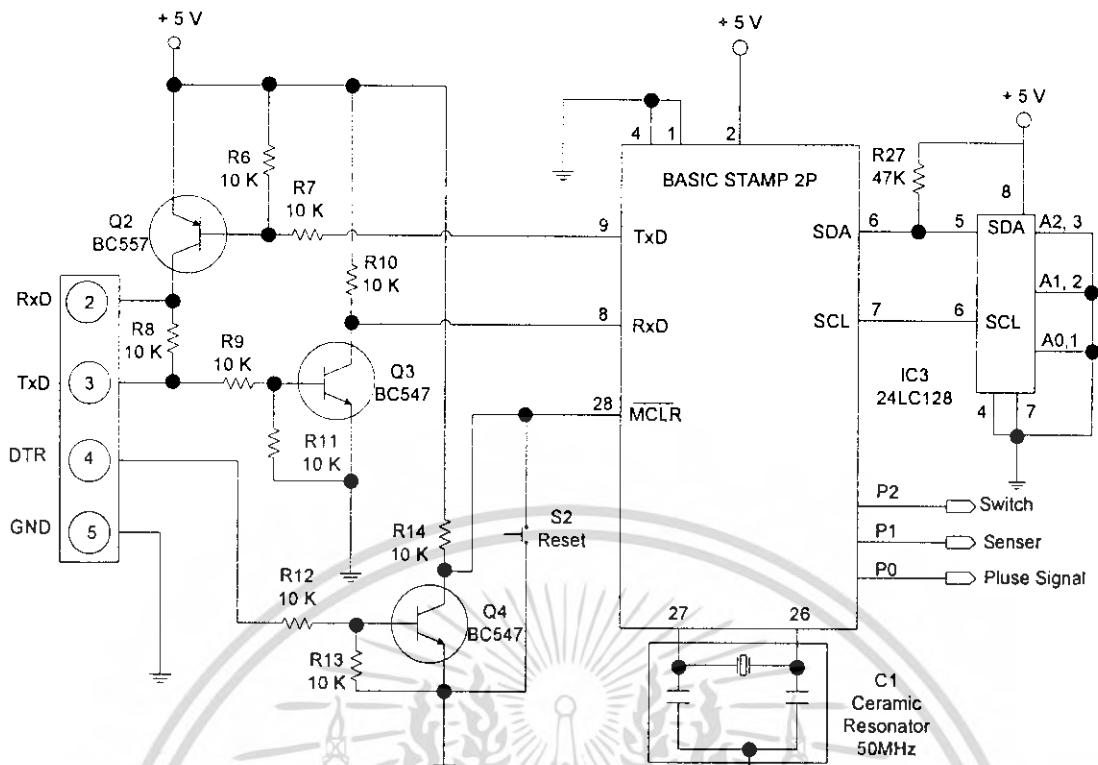


รูปที่ ก.3 ภาพวงจรเครื่องหยอดเหรียญ

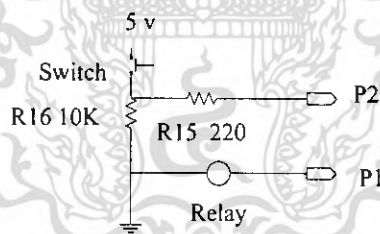
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

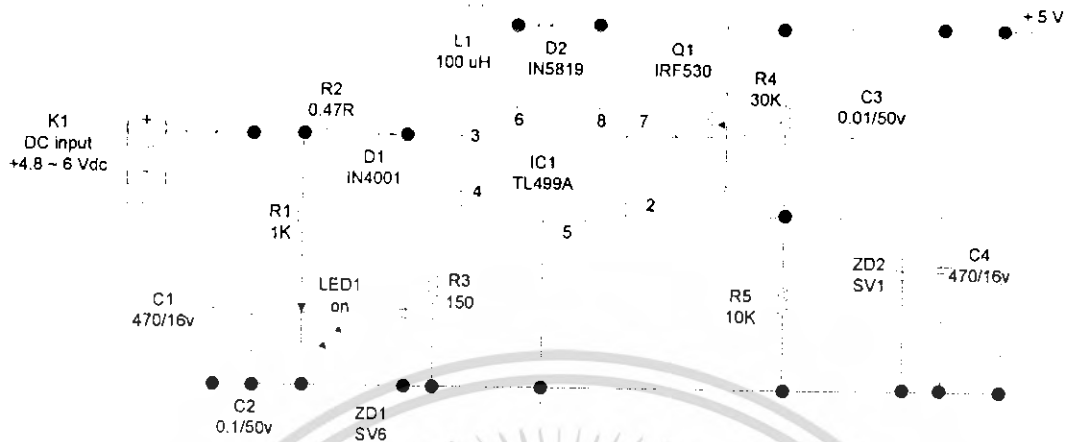


รูปที่ ข.1 วงจรสมบูรณ์ของ Stamp-Box



วงจรภายในพรินเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ข.2 วงจรควบคุมแรงดัน



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

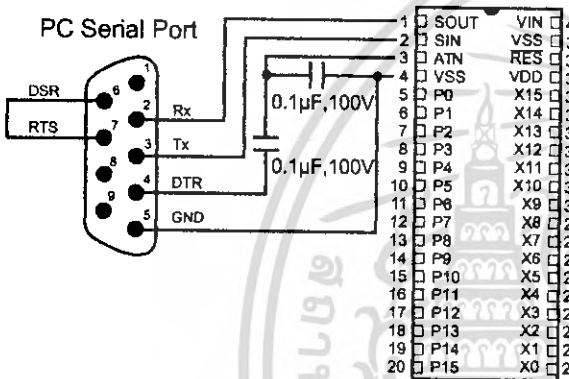


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

BS2p40-IC MODULE Rev. B (updated 5-11-04)

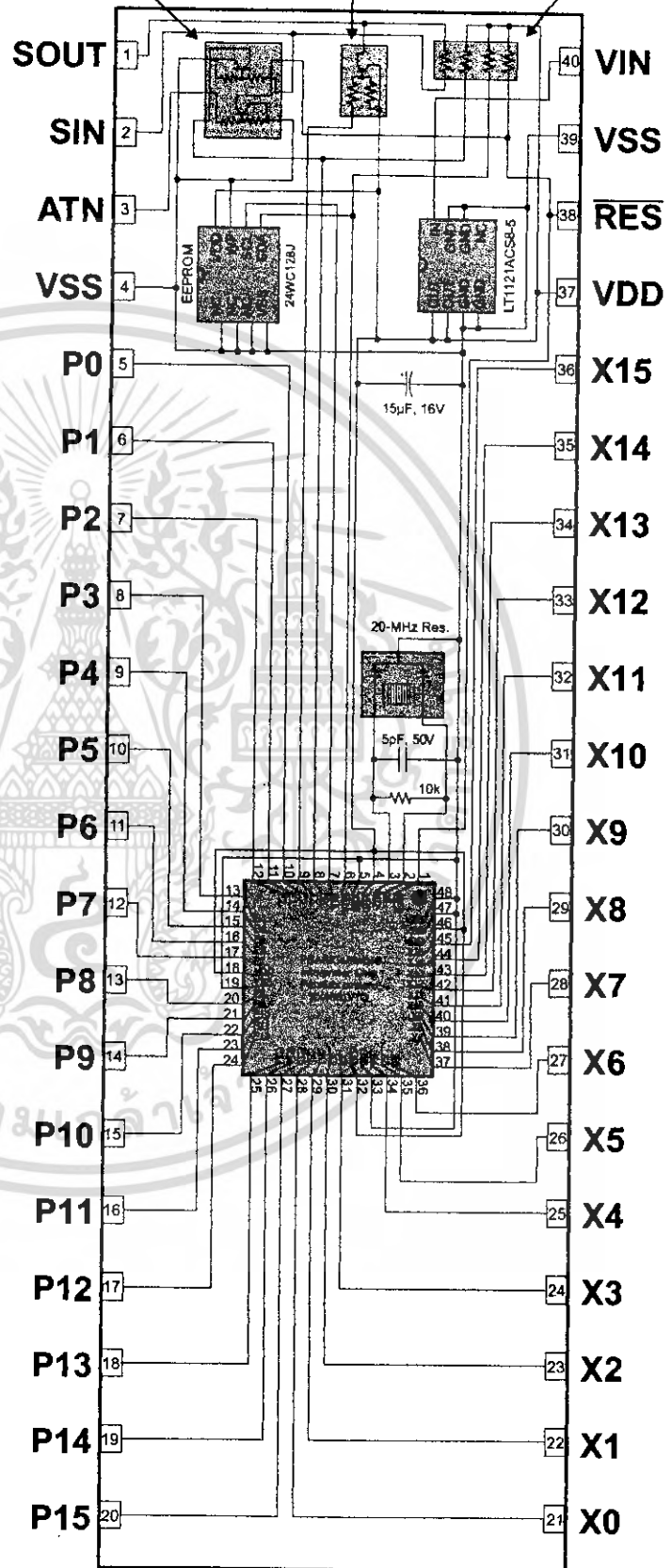
Dual NPN Trans. Pack (10K Res. 2N3904) Single PNP Trans. Pack (10K Res. & 2N3906) Quad Res. Pack (4.7K Res.)

PC - to - BS2p40-IC connection



BS2p40-IC Module

Connect DSR and RTS for automatic port detection.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Stamp Specifications (revised 04/05)

Released Products	Rev.Dx / BS1-IC		BS2-IC		BS2e-IC		BS2sx-IC	
	Package	PCB w/Proto / 14-pin SIP	24-pin DIP	24-pin DIP	24-pin DIP	24-pin DIP	24-pin DIP	24-pin DIP
Package Size (L x W x H)	2.5" x 1.5" x .5" / 1.4" x .6" x .1"	1.2" x 0.6" x 0.4"	1.2" x 0.6" x 0.4"	1.2" x 0.6" x 0.4"	1.2" x 0.6" x 0.4"	1.2" x 0.6" x 0.4"	1.2" x 0.6" x 0.4"	1.2" x 0.6" x 0.4"
Environment *	0° - 70° C (32° - 158° F) **	0° - 70° C (32° - 158° F) **	0° - 70° C (32° - 158° F) **	0° - 70° C (32° - 158° F) **	0° - 70° C (32° - 158° F)	0° - 70° C (32° - 158° F)	0° - 70° C (32° - 158° F)	0° - 70° C (32° - 158° F)
Microcontroller	Microchip PIC16C56a	Microchip PIC16C57c	Microchip PIC16C57c	Microchip PIC16C57c	Ubicom SX28AC	Ubicom SX28AC	Ubicom SX28AC	Ubicom SX28AC
Processor Speed	4 MHz	20 MHz	20 MHz	20 MHz	20 MHz	20 MHz	50 MHz	50 MHz
Program Execution Speed	~2,000 Instructions/sec.	~4,000 Instructions/sec.	~4,000 Instructions/sec.	~4,000 Instructions/sec.	~4,000 Instructions/sec.	~4,000 Instructions/sec.	~10,000 Instructions/sec.	~10,000 Instructions/sec.
RAM Size	16 Bytes (2 I/O, 14 Variable)	32 Bytes (6 I/O, 26 Variable)	32 Bytes (6 I/O, 26 Variable)	32 Bytes (6 I/O, 26 Variable)	32 Bytes (6 I/O, 26 Variable)	32 Bytes (6 I/O, 26 Variable)	32 Bytes (6 I/O, 26 Variable)	32 Bytes (6 I/O, 26 Variable)
Scratch Pad RAM	N/A	N/A	N/A	N/A	64 Bytes	64 Bytes	64 Bytes	64 Bytes
EEPROM (Program) Size	256 Bytes, ~80 instructions	2K Bytes, ~500 instructions	2K Bytes, ~500 instructions	2K Bytes, ~500 instructions	8 x 2K Bytes, ~4,000 inst.	8 x 2K Bytes, ~4,000 inst.	8 x 2K Bytes, ~4,000 inst.	8 x 2K Bytes, ~4,000 inst.
Number of I/O pins	8	16 + 2 Dedicated Serial	16 + 2 Dedicated Serial	16 + 2 Dedicated Serial	16 + 2 Dedicated Serial	16 + 2 Dedicated Serial	16 + 2 Dedicated Serial	16 + 2 Dedicated Serial
Voltage Requirements	5 - 15 vdc	5 - 15 vdc	5 - 15 vdc	5 - 15 vdc	5 - 12 vdc	5 - 12 vdc	5 - 12 vdc	5 - 12 vdc
Current Draw @ 5V	1 mA Run / 25 µA Sleep	3 mA Run / 50 µA Sleep	3 mA Run / 50 µA Sleep	3 mA Run / 50 µA Sleep	25 mA Run / 200 µA Sleep	25 mA Run / 200 µA Sleep	60 mA Run / 500 µA Sleep	60 mA Run / 500 µA Sleep
Source / Sink Current per I/O	20 mA / 25 mA	20 mA / 25 mA	20 mA / 25 mA	20 mA / 25 mA	30 mA / 30 mA	30 mA / 30 mA	30 mA / 30 mA	30 mA / 30 mA
Source / Sink Current per unit	40 mA / 50 mA	40 mA / 50 mA per 8 I/O pins	40 mA / 50 mA per 8 I/O pins	40 mA / 50 mA per 8 I/O pins	60 mA / 60 mA per 8 I/O pins	60 mA / 60 mA per 8 I/O pins	60 mA / 60 mA per 8 I/O pins	60 mA / 60 mA per 8 I/O pins
PBASIC Commands***	32	42	42	42	45	45	45	45
PC Programming Interface	Serial (w/BS1 Serial Adapter)	Serial (9600 baud)	Serial (9600 baud)	Serial (9600 baud)	Serial (9600 baud)	Serial (9600 baud)	Serial (9600 baud)	Serial (9600 baud)
Windows Text Editor	Stampw.exe (v2.1 and up)	Stampw.exe (v1.04 and up)	Stampw.exe (v1.04 and up)	Stampw.exe (v1.04 and up)	Stampw.exe (v1.096 and up)	Stampw.exe (v1.096 and up)	Stampw.exe (v1.091 and up)	Stampw.exe (v1.091 and up)

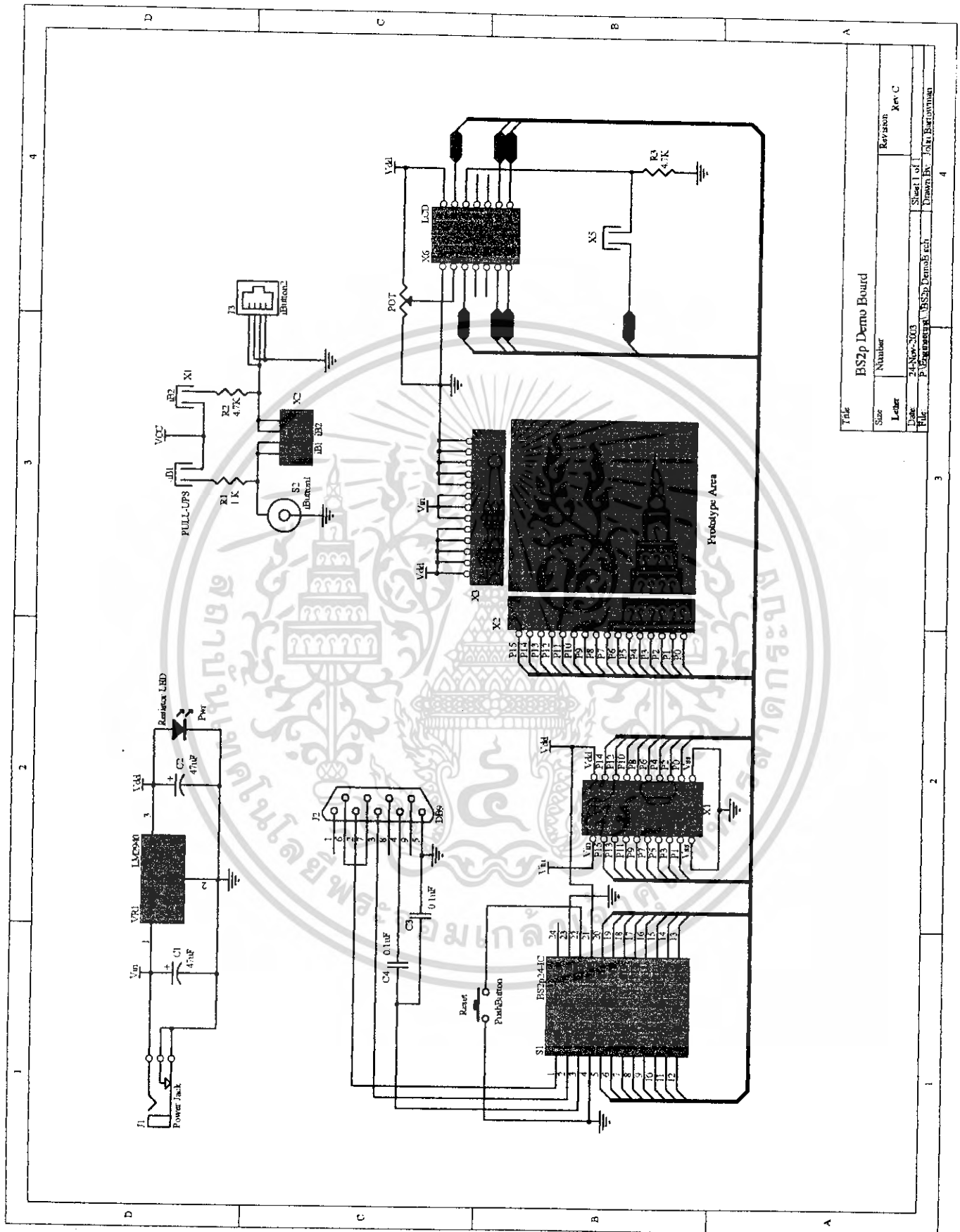
Released Products	BS2p24-IC		BS2p40-IC		BS2pe-IC		BS2px-IC		Javelin Stamp	
	Package	24-pin DIP	40-pin DIP	24-pin DIP	24-pin DIP	24-pin DIP	24-pin DIP	24-pin DIP	24-pin DIP	24-pin DIP
Package Size (L x W x H)	1.2" x 0.6" x 0.4"	1.2" x 0.6" x 0.4"	2.1" x 0.6" x 0.4"	1.2" x 0.6" x 0.4"	1.2" x 0.6" x 0.4"	1.2" x 0.6" x 0.4"	1.2" x 0.6" x 0.4"	1.2" x 0.6" x 0.4"	1.2" x 0.6" x 0.4"	1.24" x 0.60" x 0.45"
Environment *	0° - 70° C (32° - 158° F)	0° - 70° C (32° - 158° F)	0° - 70° C (32° - 158° F)	0° - 70° C (32° - 158° F)	0° - 70° C (32° - 158° F)	0° - 70° C (32° - 158° F)	0° - 70° C (32° - 158° F)	0° - 70° C (32° - 158° F)	0° - 70° C (32° - 158° F)	0° - 70° C (32° - 158° F)
Microcontroller	Ubicom SX48AC	Ubicom SX48AC	Ubicom SX48AC	Ubicom SX48AC	Ubicom SX48AC	Ubicom SX48AC	Ubicom SX48AC	Ubicom SX48AC	Ubicom SX48AC	Ubicom SX48AC
Processor Speed	20 MHz Turbo	20 MHz Turbo	20 MHz Turbo	20 MHz Turbo	8 MHz Turbo	8 MHz Turbo	32 MHz Turbo	32 MHz Turbo	25 MHz Turbo	25 MHz Turbo
Program Execution Speed	~12,000 Instructions/sec.	~12,000 Instructions/sec.	~12,000 Instructions/sec.	~12,000 Instructions/sec.	~6000/sec.	~6000/sec.	~19,000 Instructions/sec.	~19,000 Instructions/sec.	~8,500 Instructions/sec.	~8,500 Instructions/sec.
RAM Size	38 Bytes (12 I/O, 26 Variable)	38 Bytes (12 I/O, 26 Variable)	38 Bytes (12 I/O, 26 Variable)	38 Bytes (12 I/O, 26 Variable)	38 Bytes (12 I/O, 26 Variable)	38 Bytes (12 I/O, 26 Variable)	38 Bytes (12 I/O, 26 Variable)	38 Bytes (12 I/O, 26 Variable)	32768 Bytes	32768 Bytes
Scratch Pad RAM	128 Bytes	128 Bytes	128 Bytes	128 Bytes	128 Bytes	128 Bytes	128 Bytes	128 Bytes	N/A	N/A
EEPROM (Program) Size	8 x 2K Bytes, ~4,000 inst.	8 x 2K Bytes, ~4,000 inst.	8 x 2K Bytes, ~4,000 inst.	8 x 2K Bytes, ~4,000 inst.	16 x 2K Bytes (16 K for source)	16 x 2K Bytes (16 K for source)	8 x 2K Bytes, ~4,000 inst.	8 x 2K Bytes, ~4,000 inst.	32768 Bytes	32768 Bytes
Number of I/O pins	16 + 2 Dedicated Serial	16 + 2 Dedicated Serial	16 + 2 Dedicated Serial	16 + 2 Dedicated Serial	16 + 2 Dedicated Serial	16 + 2 Dedicated Serial	16 + 2 Dedicated Serial	16 + 2 Dedicated Serial	16	16
Voltage Requirements	5 - 12 vdc	5 - 12 vdc	5 - 12 vdc	5 - 12 vdc	5 - 12 vdc	5 - 12 vdc	5 - 12 vdc	5 - 12 vdc	5 - 24 vdc	5 - 24 vdc
Current Draw @ 5V	40 mA Run / 350 µA Sleep	40 mA Run / 350 µA Sleep	40 mA Run / 350 µA Sleep	40 mA Run / 350 µA Sleep	15 mA Run / 36 µA Sleep	15 mA Run / 36 µA Sleep	55 mA Run / 450 µA Sleep	55 mA Run / 450 µA Sleep	60 mA Run / No Sleep	60 mA Run / No Sleep
Source / Sink Current per I/O	30 mA / 30 mA	30 mA / 30 mA	30 mA / 30 mA	30 mA / 30 mA	30 mA / 30 mA	30 mA / 30 mA	30 mA / 30 mA	30 mA / 30 mA	30 mA / 30 mA	30 mA / 30 mA
Source / Sink Current per unit	60 mA / 60 mA per 8 I/O pins	60 mA / 60 mA per 8 I/O pins	60 mA / 60 mA per 8 I/O pins	60 mA / 60 mA per 8 I/O pins	60 mA / 60 mA per 8 I/O pins	60 mA / 60 mA per 8 I/O pins	60 mA / 60 mA per 8 I/O pins	60 mA / 60 mA per 8 I/O pins	60 mA / 60 mA per 8 I/O pins	60 mA / 60 mA per 8 I/O pins
PBASIC Commands***	61	61	61	61	61	61	63	63	0 (Java)	0 (Java)
PC Programming Interface	Serial (9600 baud)	Serial (9600 baud)	Serial (9600 baud)	Serial (9600 baud)	Serial (9600 baud)	Serial (9600 baud)	Serial (19200 baud)	Serial (19200 baud)	Serial (28800 baud)	Serial (28800 baud)
Windows Text Editor	Stampw.exe (v1.1 and up)	Stampw.exe (v1.1 and up)	Stampw.exe (v1.1 and up)	Stampw.exe (v1.1 and up)	Stampw.exe (v1.33 and up)	Stampw.exe (v1.33 and up)	Stampw.exe (v2.2 and up)	Stampw.exe (v2.2 and up)	Stampw.exe (v2.2 and up)	Stampw.exe (v2.2 and up)

* 70% Non-Condensing Humidity

** Industrial Models Available, -40° - 85° C (-40° - 185° F). Contact Parallax Sales for information.

*** Using PBASIC 2.5 for BS2-type models.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



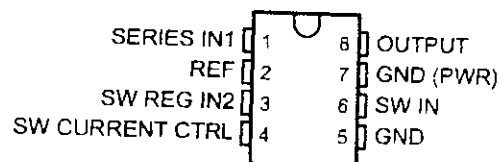
Title	ES2p Demo Board		
Size	Number	Revision	Rev C
Letter			
Date	24-Nov-2003	Sheet 1 of 1	
File	P:\BSP\BSP Demo\BSP Demo.rch	Drawn By	Jai Buranawan

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TL499A WIDE-RANGE POWER-SUPPLY CONTROLLERS

SLVS029F – JANUARY 1984 – REVISED MAY 2000

- Internal Series-Pass and Step-Up Switching Regulator
- Output Adjustable From 2.9 V to 30 V
- 1-V to 10-V Input for Switching Regulator
- 4.5-V to 32-V Input for Series Regulator
- Externally Controlled Switching Current
- No External Rectifier Required

P OR PS PACKAGE
(TOP VIEW)

description

The TL499A is an integrated circuit designed to provide a wide range of adjustable regulated supply voltages. The regulated output voltage can be varied from 2.9 V to 30 V by adjusting two external resistors. When the TL499A is ac-coupled to line power through a step-down transformer, it operates as a series dc voltage regulator to maintain the regulated output voltage. With the addition of a battery from 1.1 V to 10 V, an inductor, a filter capacitor, and two resistors, the TL499A operates as a step-up switching regulator during an ac-line failure.

The adjustable regulated output voltage makes the TL499A useful for a wide range of applications. Providing backup power during an ac-line failure makes the TL499A extremely useful in microprocessor memory applications.

The TL499AC is characterized for operation from -20°C to 85°C .

AVAILABLE OPTIONS

TA	PLASTIC DIP (P)	PLASTIC SMALL-OUTLINE (PS)	CHIP FORM (Y)
-20°C to 85°C	TL499ACP	TL499ACPS	TL499AY

The PS package is available taped and reeled. Add the suffix R to device type (e.g., TL499ACPSR). Chip forms are tested at 25°C .



Please be aware that an important notice concerning availability, standard warranty, and use in critical applications of Texas Instruments semiconductor products and disclaimers thereto appears at the end of this data sheet.

PRODUCTION DATA information is current as of publication date. Products conform to specifications per the terms of Texas Instruments standard warranty. Production processing does not necessarily include testing of all parameters.

**TEXAS
INSTRUMENTS**

POST OFFICE BOX 655303 • DALLAS, TEXAS 75265

Copyright © 2000, Texas Instruments Incorporated

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TL499A WIDE-RANGE POWER-SUPPLY CONTROLLERS

SLVS029F - JANUARY 1984 - REVISED MAY 2000

APPLICATION INFORMATION

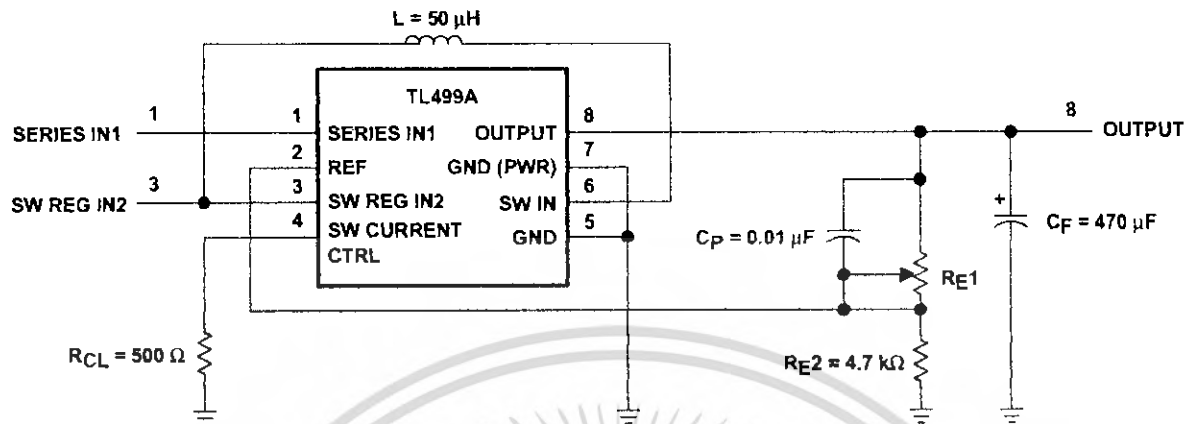


Figure 1. TL499A Basic Configuration

Table 1. Maximum Output Current vs Input and Output Voltages
for Step-Up Switching Regulator With $R_{CL} = 150 \Omega$

OUTPUT VOLTAGE (V)	SWITCHING REGULATOR INPUT VOLTAGE (SW REG IN2) (V)										
	1.1	1.2	1.3	1.5	1.7	2	2.5	3	5	6	9
	OUTPUT CURRENT (mA)										
30										65	90
25									50	80	100
20						20	25	30	80	100	100
15				15	20	30	45	55	100	100	100
12	10	15	20	25	30	40	55	70	100	100	100
10	15	20	25	30	35	45	65	80	100	100	
9	20	25	25	35	40	50	70	90	100	100	
6	30	35	40	45	55	75	95	100			
5	35	40	45	55	70	85	100	100	Circuit of Figure 1, except: $R_{CL} = 150 \Omega$ $C_F = 330 \mu F$ $C_P = 0.1 \mu F$		
4.5	35	45	50	60	75	95	100	100†			
3	55	65†	75†	95†	100†						
2.9	60†	70†	75†	100†	100†						

† The difference between the output and input voltage for these combinations is greater than the minimum output-to-input differential-voltage specification at 70°C (1.2 V), but less than the minimum at 85°C (1.9 V).

 **TEXAS
INSTRUMENTS**

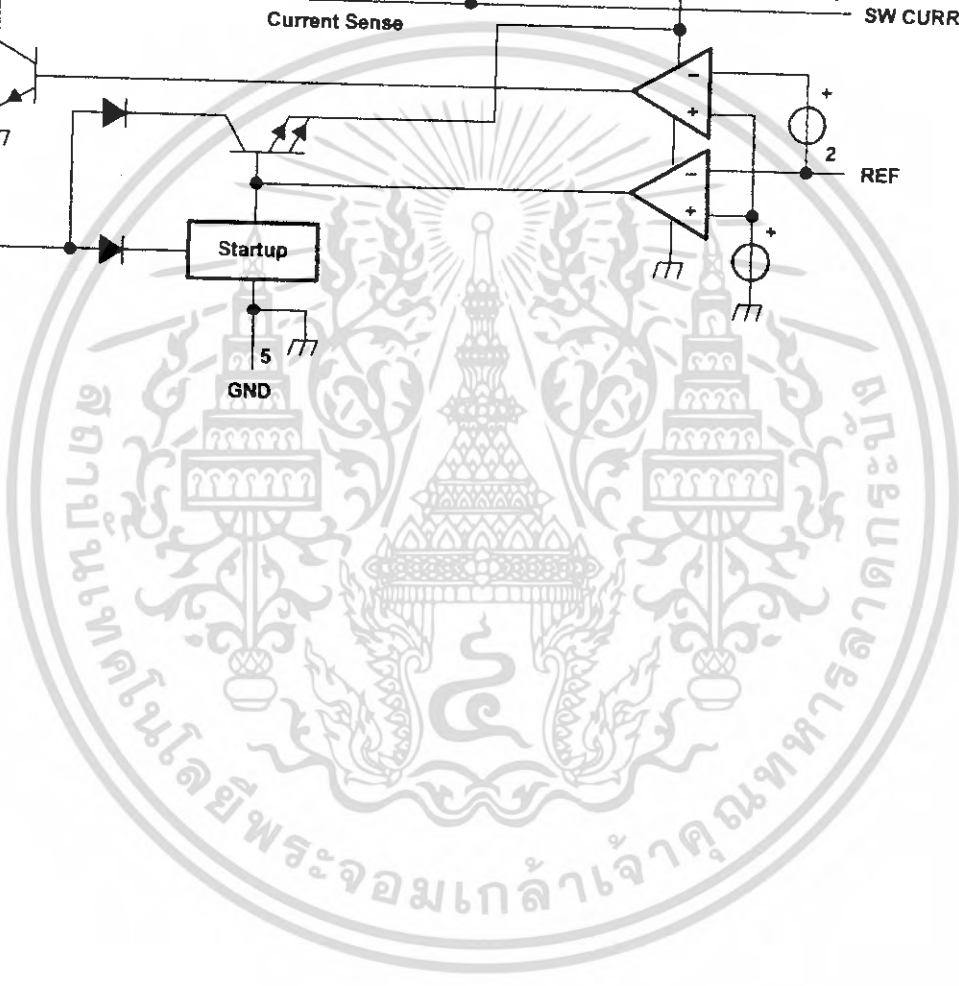
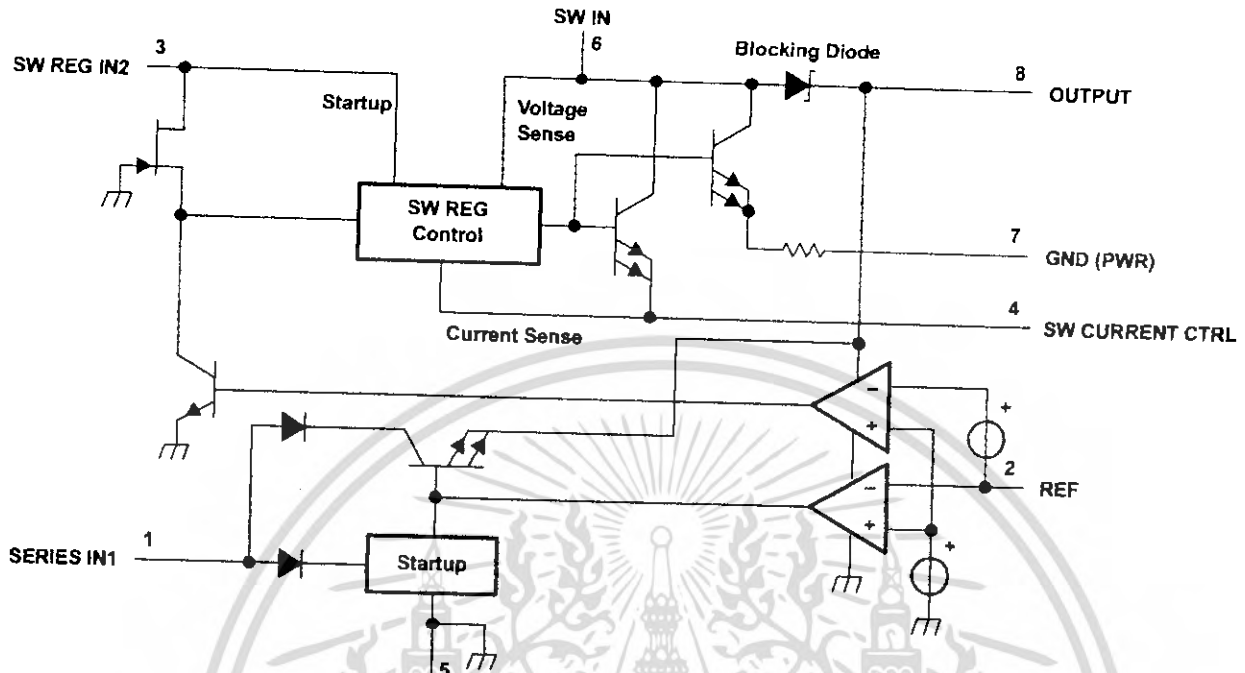
POST OFFICE BOX 655303 • DALLAS, TEXAS 75265.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TL499A WIDE-RANGE POWER-SUPPLY CONTROLLERS

SLVS029F – JANUARY 1984 – REVISED MAY 2000

functional block diagram



POST OFFICE BOX 655303 • DALLAS, TEXAS 75265

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

APPLICATION INFORMATION

Table 2. Maximum Output Current vs Input and Output Voltages for Step-Up Switching Regulator With $R_{CL} = 200 \Omega$

OUTPUT VOLTAGE (V)	SWITCHING REGULATOR INPUT VOLTAGE (SW REG IN2) (V)										
	1.1	1.2	1.3	1.5	1.7	2	2.5	3	5	6	9
	OUTPUT CURRENT (mA)										
30										50	100
25									50	70	100
20						15	25	30	70	90	100
15				10	15	25	35	45	90	100	100
12	10	10	15	20	25	35	45	60	100	100	100
10	15	20	20	25	30	40	55	70	100	100	
9	20	20	25	30	35	45	60	80	100		
6	25	30	35	45	50	65	90	100			
5	30	35	40	55	60	75	100	100			
4.5	35	40	45	55	65	85	100	100†	Circuit of Figure 1, except: $R_{CL} = 200 \Omega$ $C_F = 330 \mu F$ $C_P = 0.1 \mu F$		
3	50	55†	65†	80†	90†						
2.9	50†	60†	65†	85†	100†						

† The difference between the output and input voltage for these combinations is greater than the minimum output-to-input differential-voltage specification at 70°C (1.2 V), but less than the minimum at 85°C (1.9 V).

Table 3. Maximum Output Current vs Input and Output Voltages for Step-Up Switching Regulator With $R_{CL} = 300 \Omega$

OUTPUT VOLTAGE (V)	SWITCHING REGULATOR INPUT VOLTAGE (SW REG IN2) (V)										
	1.1	1.2	1.3	1.5	1.7	2	2.5	3	5	6	9
	OUTPUT CURRENT (mA)										
30										40	70
25									40	55	100
20						10	15	20	55	70	100
15				10	10	20	30	35	75	95	100
12	10	10	10	15	20	25	35	45	95	100	100
10	15	15	15	20	25	30	45	55	100	100	
9	15	15	20	25	30	35	50	60	100	100	
6	25	25	30	35	45	55	70	90			
5	30	30	35	45	50	65	85	100			
4.5	30	35	40	45	55	70	95	100†	Circuit of Figure 1, except: $R_{CL} = 300 \Omega$ $C_F = 330 \mu F$ $C_P = 0.1 \mu F$		
3	45	50†	55†	70†	90†						
2.9	45†	50†	60†	75†	95†						

† The difference between the output and input voltage for these combinations is greater than the minimum output-to-input differential-voltage specification at 70°C (1.2 V), but less than the minimum at 85°C (1.9 V).



POST OFFICE BOX 655303 • DALLAS, TEXAS 75265

TL499A

WIDE-RANGE POWER-SUPPLY CONTROLLERS

SLVS029F – JANUARY 1984 – REVISED MAY 2000

electrical characteristics over recommended operating conditions (unless otherwise noted)

PARAMETER	TEST CONDITIONS	TL499AC			UNIT	
		MIN	TYP	MAX		
Voltage deviation (see Note 5)			20	30	mV/V	
Dropout voltage	Switching regulator	$T_A = -20^\circ\text{C to } 70^\circ\text{C}$			1.2	V
	Series regulator	$V_{I1} = 15\text{ V}, I_O = 50\text{ mA}$			1.9	
Reference voltage (internal)	$V_{I2} = 5\text{ V}, V_O = 3\text{ V}, I_O = 1\text{ mA}$	1.2	1.26	1.32	V	
Reference-voltage change with temperature			5	10	mV/V	
Output regulation (of reference voltage)	$I_O = 1\text{ mA to } 50\text{ mA}$		10	30	mV/V	
Output current (see Figure 1)	Switching regulator	$V_{I2} = 1.1\text{ V}, V_O = 12\text{ V}, R_{CL} = 150\ \Omega, T_A = 25^\circ\text{C}$	10		mA	
		$V_{I2} = 1.5\text{ V}, V_O = 15\text{ V}, R_{CL} = 150\ \Omega, T_A = 25^\circ\text{C}$	15			
		$V_{I2} = 6\text{ V}, V_O = 30\text{ V}, R_{CL} = 150\ \Omega, T_A = 25^\circ\text{C}$	65			
Series regulator		100				
Standby current	Switching regulator	$V_{I2} = 3\text{ V}, V_O = 9\text{ V}, T_A = 25^\circ\text{C}$	15	80	μA	
	Series regulator	$V_{I1} = 15\text{ V}, V_O = 9\text{ V}, R_{E2} = 4.7\text{ k}\Omega$	0.8	1.2	mA	

NOTE 5: Voltage deviation is the output voltage difference that occurs in a change from series regulation to switching regulation:
 Voltage deviation = $V_{O(\text{series regulation})} - V_{O(\text{switching regulation})}$

electrical characteristics over recommended operating conditions, $T_A = 25^\circ\text{C}$ (unless otherwise noted)

PARAMETER	TEST CONDITIONS	TL499AY			UNIT	
		MIN	TYP	MAX		
Voltage deviation (see Note 5)			20	30	mV/V	
Dropout voltage	Switching regulator	$T_A = -20^\circ\text{C to } 70^\circ\text{C}$			1.2	V
	Series regulator	$T_A = -20^\circ\text{C to } 85^\circ\text{C}$			1.9	
Reference voltage (internal)	$V_{I1} = 15\text{ V}, I_O = 50\text{ mA}$			1.8	V	
Reference voltage (internal)	$V_{I2} = 5\text{ V}, V_O = 3\text{ V}, I_O = 1\text{ mA}$	1.2	1.26	1.32	V	
Reference-voltage change with temperature	$T_A = -20^\circ\text{C to } 85^\circ\text{C}$		5	10	mV/V	
Output regulation (of reference voltage)	$I_O = 1\text{ mA to } 50\text{ mA}$		10	30	mV/V	
Output current (see Figure 1)	Switching regulator	$V_{I2} = 1.1\text{ V}, V_O = 12\text{ V}, R_{CL} = 150\ \Omega$	10		mA	
		$V_{I2} = 1.5\text{ V}, V_O = 15\text{ V}, R_{CL} = 150\ \Omega$	15			
		$V_{I2} = 6\text{ V}, V_O = 30\text{ V}, R_{CL} = 150\ \Omega$	65			
Series regulator		100				
Standby current	Switching regulator	$V_{I2} = 3\text{ V}, V_O = 9\text{ V}$	15	80	μA	
	Series regulator	$V_{I1} = 15\text{ V}, V_O = 9\text{ V}, R_{E2} = 4.7\text{ k}\Omega$	0.8	1.2	mA	

NOTE 5: Voltage deviation is the output voltage difference that occurs in a change from series regulation to switching regulation:
 Voltage deviation = $V_{O(\text{series regulation})} - V_{O(\text{switching regulation})}$



POST OFFICE BOX 655303 • DALLAS, TEXAS 75265



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้าเมนู

Option Explicit

Dim FrameUse, Ch, Opt As Integer

Private Sub Frame1_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)

Dim ReferFrame As Integer

Call ClearFrameColor

End Sub

Private Sub Frame2_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)

Call ClearFrameColor

Frame2.BackColor = &HFFFFFFC0

Label3.BackColor = &HFFFFFFC0

End Sub

Private Sub Image2_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)

Call ClearFrameColor

Frame2.BackColor = &HFFFFFFC0

Label3.BackColor = &HFFFFFFC0

End Sub

Private Sub Label3_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)

Call ClearFrameColor

Frame2.BackColor = &HFFFFFFC0

Label3.BackColor = &HFFFFFFC0

End Sub

Private Sub Frame3_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)

Call ClearFrameColor

Frame3.BackColor = &HFFFFFFC0

Label4.BackColor = &HFFFFFFC0

End Sub

Private Sub Image3_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)

Call ClearFrameColor

Frame3.BackColor = &HFFFFFFC0

Label4.BackColor = &HFFFFFFC0

End Sub

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Private Sub Label4_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
    Call ClearFrameColor
        Frame3.BackColor = &HFFFFFF0
        Label4.BackColor = &HFFFFFF0
    End Sub
Private Sub Frame4_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
    Call ClearFrameColor
        Frame4.BackColor = &HFFFFFF0
        Label5.BackColor = &HFFFFFF0
    End Sub
Private Sub Image4_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
    Call ClearFrameColor
        Frame4.BackColor = &HFFFFFF0
        Label5.BackColor = &HFFFFFF0
    End Sub
Private Sub Label5_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
    Call ClearFrameColor
        Frame4.BackColor = &HFFFFFF0
        Label5.BackColor = &HFFFFFF0
    End Sub
Public Sub ClearFrameColor()
    Frame3.BackColor = &HFFFFFFF
    Frame4.BackColor = &HFFFFFFF
    Frame2.BackColor = &HFFFFFFF
    Frame1.BackColor = &HFFFFFFF
    Label3.BackColor = &HFFFFFFF
    Label4.BackColor = &HFFFFFFF
    Label5.BackColor = &HFFFFFFF
    End Sub
Private Sub Frame2_Click()
    Unload Form3
    Form2.Show

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

End Sub
Private Sub Image2_Click()
Unload Form3
Form2.Show
End Sub
Private Sub Label3_Click()
Unload Form3
Form2.Show
End Sub
Private Sub Frame3_Click()
Unload Form3
Form1.Show
End Sub
Private Sub Image3_Click()
Unload Form3
Form1.Show
End Sub
Private Sub Label4_Click()
Unload Form3
Form1.Show
End Sub
Private Sub Frame4_Click()
End
End Sub
Private Sub Image4_Click()
End
End Sub
Private Sub Label5_Click()
End
End Sub

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบการพิมพ์

```

Public data As String
Public recI As Integer
Public sen As Integer
Dim DB As Database
Dim RS As Recordset
Dim sql As String
Dim sw As Integer
Dim MyTime As String
Dim MyMonth As String
Dim MyYear As String
Dim MyDate As Date
Public coin As Integer
Private Sub Command1_Click()
End Sub
Private Sub Form_Load()
MSComm1.CommPort = 1
MSComm1.Settings = "9600,n,8,1"
MSComm1.PortOpen = True
Timer1.Interval = 100
Timer1.Enabled = True
End Sub
Private Sub Image1_Click()
Dim da As Integer
If coin > 0 Then
da = coin
sen = Fix(da / 2)   'เปลี่ยนราคา
If sen > 0 Then
MSComm1.Output = Chr(sen)
End If
End If
End Sub

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

MyDate = Format(Now, "DDDDDD")
MyTime = Format(Now, "Short Time")
Set DB = DBEngine.Workspaces(0).OpenDatabase(App.Path & "\vb6.mdb")
sql = "insert into table1 (Cost,DDate,TTime)"
sql = sql & "values( "" & coin & "", "" & MyDate & "", "" & MyTime & "" )"
DB.Execute (sql)
MyMonth = Format(Now, "m/yy")
Set DB = DBEngine.Workspaces(0).OpenDatabase(App.Path & "\vb6.mdb")
sql = "insert into Table2 (Cost,MMonth)"
sql = sql & "values( "" & coin & "", "" & MyMonth & "" )"
DB.Execute (sql)
MyYear = Format(Now, "yy")
Set DB = DBEngine.Workspaces(0).OpenDatabase(App.Path & "\vb6.mdb")
sql = "insert into Table3 (Cost,Yyear)"
sql = sql & "values( "" & coin & "", "" & MyYear & "" )"
DB.Execute (sql)
sw = 0
coin = 0
End Sub

Private Sub Image2_Click()
MSComm1.Output = Chr(255)
End Sub

Private Sub Timer1_Timer()
Dim rec1 As Integer
Dim rec As Integer

If sw = 0 Then
'On Error Resume Next
MSComm1.DTREnable = False

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

MSComm1.DTREnable = True
MSComm1.InputLen = 1024
data = MSComm1.Input
rec1 = Val(data)
'rec = rec + data
coin = coin + rec1
Label7.Caption = coin
Label5.Caption = Fix(coin / 2)   เปลี่ยนราคา
Else
    Call prin
    End If
End Sub
Sub prin()
sw = 1
End
End Sub

Private Sub Image3_Click()
Unload Form2
Form3.Show
End Sub

Private Sub Label3_Click()
Unload Form2
Form3.Show
End Sub

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบฐานข้อมูล

```

Option Explicit
Dim DB As Database
Dim RS As Recordset
Dim FrameUse, Ch, Opt As Integer
Dim sql, Tmp As String
Dim ITEMS As ListItem
Dim MyTime As String
Dim MyMonth As String
Dim MyYear As String
Dim MyDate As Date
Private Sub cmd2_Click()
End
End Sub

Private Sub Frame1_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
    Dim ReferFrame As Integer
    Call ClearFrameColor
End Sub

Private Sub Frame4_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
    Call ClearFrameColor
    Frame4.BackColor = &HFFFFFFC0
    Label1.BackColor = &HFFFFFFC0
End Sub

Private Sub Image2_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
    Call ClearFrameColor
    Frame4.BackColor = &HFFFFFFC0
    Label1.BackColor = &HFFFFFFC0
End Sub

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Private Sub Label1_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
    Call ClearFrameColor
    Frame4.BackColor = &HFFFFFFC0
    Label1.BackColor = &HFFFFFFC0
End Sub
```

```
Private Sub Frame7_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
    Call ClearFrameColor
    Frame7.BackColor = &HFFFFFFC0
    Label3.BackColor = &HFFFFFFC0
End Sub
```

```
Private Sub Image3_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
    Call ClearFrameColor
    Frame7.BackColor = &HFFFFFFC0
    Label3.BackColor = &HFFFFFFC0
End Sub
```

```
Private Sub Label3_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
    Call ClearFrameColor
    Frame7.BackColor = &HFFFFFFC0
    Label3.BackColor = &HFFFFFFC0
End Sub
```

```
Private Sub Frame8_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
    Call ClearFrameColor
    Frame8.BackColor = &HFFFFFFC0
    Label4.BackColor = &HFFFFFFC0
End Sub
```

```
Private Sub Image4_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
    Call ClearFrameColor
    Frame8.BackColor = &HFFFFFFC0
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Label4.BackColor = &HFFFFFFC0
End Sub

Private Sub Label4_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
Call ClearFrameColor
Frame8.BackColor = &HFFFFFFC0
Label4.BackColor = &HFFFFFFC0
End Sub

Private Sub Frame2_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
Call ClearFrameColor
Frame2.BackColor = &HFFFFFFC0
Label7.BackColor = &HFFFFFFC0
End Sub

Private Sub Image1_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
Call ClearFrameColor
Frame2.BackColor = &HFFFFFFC0
Label7.BackColor = &HFFFFFFC0
End Sub

Private Sub Label7_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
Call ClearFrameColor
Frame2.BackColor = &HFFFFFFC0
Label7.BackColor = &HFFFFFFC0
End Sub

Private Sub Frame4_Click()
Call ClearFrameForm
Frame5.Visible = True
FrameUse = 5
End Sub

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Private Sub Frame7_Click()
    Call ClearFrameForm
        Frame6.Visible = True
        FrameUse = 7
End Sub

```

```

Private Sub Frame8_Click()
' Call ClearFrameForm
    ' Frame10.Visible = True
    ' FrameUse = 8
End Sub

```

```

Private Sub Image2_Click()
    Call ClearFrameForm
        Frame5.Visible = True
        FrameUse = 5
End Sub

```

```

Private Sub Image3_Click()
    Call ClearFrameForm
        Frame6.Visible = True
        FrameUse = 7
End Sub

```

```

Private Sub Image4_Click()
' Call ClearFrameForm
    ' Frame10.Visible = True
    ' FrameUse = 8
End Sub

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Private Sub Label2_Click()
```

```
    Call ClearFrameForm
```

```
    Frame2.Visible = True
```

```
    FrameUse = 1
```

```
End Sub
```

```
Private Sub Label1_Click()
```

```
    Call ClearFrameForm
```

```
    Frame5.Visible = True
```

```
    FrameUse = 5
```

```
End Sub
```

```
Private Sub Label3_Click()
```

```
    Call ClearFrameForm
```

```
    Frame6.Visible = True
```

```
    FrameUse = 7
```

```
End Sub
```

```
Private Sub Frame2_Click()
```

```
    Form3.Show
```

```
    Unload Form1
```

```
End Sub
```

```
Private Sub Image1_Click()
```

```
    Form3.Show
```

```
    Unload Form1
```

```
End Sub
```

```
Private Sub Label7_Click()
```

```
    Form3.Show
```

```
    Unload Form1
```

```
End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Public Sub ClearFrameForm()
```

```
    Frame5.Visible = False
```

```
    Frame6.Visible = False
```

```
End Sub
```

```
Public Sub ClearFrameColor()
```

```
    Frame2.BackColor = &HFFFFFF
```

```
    Frame4.BackColor = &HFFFFFF
```

```
    Frame5.BackColor = &HFFFFFF
```

```
    Frame6.BackColor = &HFFFFFF
```

```
    Frame7.BackColor = &HFFFFFF
```

```
    Frame8.BackColor = &HFFFFFF
```

```
    Label1.BackColor = &HFFFFFF
```

```
    Label3.BackColor = &HFFFFFF
```

```
    Label4.BackColor = &HFFFFFF
```

```
    Label7.BackColor = &HFFFFFF
```

```
End Sub
```

```
Private Sub OLE1_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
```

```
    Call ClearFrameColor
```

```
    Frame8.BackColor = &HFFFC0
```

```
    Label4.BackColor = &HFFFC0
```

```
End Sub
```

```
Private Sub Option1_Click()
```

```
    If Option1.Value = True Then
```

```
        Text2.Text = ""
```

```
        ListView1.ListItems.Clear
```

```
        ListView1.ColumnHeaders.Clear
```

```
        ListView1.ColumnHeaders.Add , , "ID"
```

```
        ListView1.ColumnHeaders.Add , , "Cost"
```

```
        ListView1.ColumnHeaders.Add , , "Date"
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        ListView1.ColumnHeaders.Add , , "Time"
    End If
End Sub

Private Sub Option2_Click()
    If Option2.Value = True Then
        Text2.Text = ""
        ListView1.ListItems.Clear
        ListView1.ColumnHeaders.Clear
        ListView1.ColumnHeaders.Add , , "' ", (ListView1.Width / 5) - 40
        ListView1.ColumnHeaders.Add , , "' ", (ListView1.Width / 5) - 40
        ListView1.ColumnHeaders.Add , , "' ", (ListView1.Width / 5) - 40
    End If
End Sub

Private Sub Text2_Change()
    If Text2.Text = "" Then
        ListView1.ListItems.Clear
    End If
    Tmp = Text2.Text
    If Option1.Value = True Then
        Set DB = DBEngine.Workspaces(0).OpenDatabase(App.Path & "\vb6.mdb")
        sql = "select * from Table1 where ID like '*' & Tmp & '*' or "
        sql = sql & "Cost like '*' & Tmp & '*' or DDate like '*' & Tmp & '*' or TTime like '*' & "
        Tmp & "*" "
        Set RS = DB.OpenRecordsset(sql)
    End If
    ListView1.ListItems.Clear
    While Not RS.EOF
        'Set ITEMS = ListView1.ListItems.Add( , , "11111111")
        Set ITEMS = ListView1.ListItems.Add( , , CStr(RS.Fields(0)))
        ITEMS.SubItems(1) = RS.Fields(1)
    End While
End Sub

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

ITEMS.SubItems(2) = RS.Fields(2)
ITEMS.SubItems(3) = RS.Fields(3)
RS.MoveNext
Wend
End Sub
Private Sub Option3_Click()
    If Option3.Value = True Then
        ListView2.ListItems.Clear
        ListView2.ColumnHeaders.Clear
        ListView2.ColumnHeaders.Add , , "Date"
        ListView2.ColumnHeaders.Add , , "Cost"
    End If
    Set DB = DBEngine.Workspaces(0).OpenDatabase(App.Path & "\vb6.mdb")
    sql = "select DDate,sum(Cost) from Table1 GROUP BY DDate ORDER BY DDate "
    Set RS = DB.OpenRecordset(sql)
    ListView2.ListItems.Clear
    While Not RS.EOF
        Set ITEMS = ListView2.ListItems.Add(, , CStr(RS.Fields(0)))
        ITEMS.SubItems(1) = RS.Fields(1)
    RS.MoveNext
    Wend
End Sub
Private Sub Option4_Click()
    If Option4.Value = True Then
        ListView2.ListItems.Clear
        ListView2.ColumnHeaders.Clear
        ListView2.ColumnHeaders.Add , , "" ' , (ListView1.Width / 5) - 40
        ListView2.ColumnHeaders.Add , , "" ' , (ListView1.Width / 5) - 40
    End If
End Sub
Private Sub Option5_Click()
    If Option5.Value = True Then

```

```

ListView2.ListItems.Clear

ListView2.ColumnHeaders.Clear

ListView2.ColumnHeaders.Add , , "Month"

ListView2.ColumnHeaders.Add , , "Cost"

End If

Set DB = DBEngine.Workspaces(0).OpenDatabase(App.Path & "\vb6.mdb")

sql = "select Mmonth,sum(Cost) from Table2 GROUP BY Mmonth ORDER BY Mmonth "

Set RS = DB.OpenRecordset(sql)

ListView2.ListItems.Clear

While Not RS.EOF

Set ITEMS = ListView2.ListItems.Add(, , CStr(RS.Fields(0)))

ITEMS.SubItems(1) = RS.Fields(1)

RS.MoveNext

Wend

End Sub

Private Sub Option6_Click()

If Option6.Value = True Then

ListView2.ListItems.Clear

ListView2.ColumnHeaders.Clear

ListView2.ColumnHeaders.Add , , "Year"

ListView2.ColumnHeaders.Add , , "Cost"

End If

Set DB = DBEngine.Workspaces(0).OpenDatabase(App.Path & "\vb6.mdb")

sql = "select Yyear,sum(Cost) from Table3 GROUP BY Yyear ORDER BY Yyear "

Set RS = DB.OpenRecordset(sql)

ListView2.ListItems.Clear

While Not RS.EOF

Set ITEMS = ListView2.ListItems.Add(, , CStr(RS.Fields(0)))

ITEMS.SubItems(1) = RS.Fields(1)

RS.MoveNext

Wend

End Sub

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เบสิกสเตมปี 2p

```

'{$STAMP BS2P}
'{$SPORT COM1}
serData VAR Byte
Cycles VAR Byte
cycles =0
pop VAR Byte
bynwrk VAR Byte
bynwrk=0
swt VAR Byte
DIR2=1
start:
IF IN0 = 0 THEN loop
BUTTON 1,1,255,250,bynwrk,1,ligh 'sw paper
SERIN 16, 16624,100,start,[ serdata]
IF serdata=225 THEN setprin 'printet erro
FOR pop=1 TO serdata 'add paper
cycles=cycles+1
NEXT
IF swt=0 THEN swith
GOTO start
ligh:
cycles=cycles-1
IF cycles = 0 THEN setpro
GOTO start
setpro:
swt=0
OUT2=1
PAUSE 300
OUT2=0
GOTO start
Loop:

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

COUNT 0,6000 ,Cycles
SEROUT 16, 16624, [DEC Cycles,CR] 'sen coin to computer
GOTO start
with:
swt=1
OUT2=1
PAUSE 300
OUT2=0
GOTO start
setprin:
swt=1
OUT2=1
PAUSE 3000
OUT2=0
OUT2=1
PAUSE 2000
OUT2=0
GOTO start

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาคผนวก จ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คู่มือการใช้งาน

1. เมื่อทำการหยุดเหรียญจะสามารถทราบจำนวนหน้ากระดาษที่สามารถพิมพ์ได้
2. เปิดไฟล์ที่ต้องการจะพิมพ์
3. ทำการสั่งพิมพ์
4. กรณีที่เครื่องพริ้นเตอร์ไม่ทำงานหรือทำงานผิดพลาด
 - 4.1 สั่งยกเลิกการพิมพ์งาน
 - 4.2 คลิกที่ปุ่ม ResetPrinter



Restart Printer

- 4.3 รอจนกว่าเครื่องพริ้นเตอร์จะทำการเปิดและปิดตัวเอง
- 4.4 ทำการพิมพ์งานได้ตามปกติ
5. การค้นหาข้อมูล
 - 5.1 ทำการคลิกที่ วัน/เดือน/ปี
 - 5.2 ใส่ วัน/เดือน/ปี ที่เราต้องการค้นหาลงใน Text box
6. การค้นหารายได้ที่ได้รับ
 - 6.1 คลิกที่รายได้ที่ได้รับ
 - 6.2 ทำการเลือกรายได้แต่ละวัน, เดือน และปีที่ต้องการทราบ
7. กราฟรายได้แต่ละเดือน
 - 7.1 ทำการคลิกที่กราฟที่จะปรากฏกราฟโชว์ขึ้นมา