สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

เครื่องหยอดเหรียญ

COINS COUNTER

นาย เฉลิมชัย ลาปัน รหัสประจำตัว 41013284
นาย ณัฐกุล บุญญะอำไพวงศ์ รหัสประจำตัว 41013287

เลขหมู่.....
เลขทะเบียน 42663
วัน, เดือน, ปี - 6 มิ.ย. 2545

.b.....
.i.....

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาอุตสาหกรรมศาสตรบัณฑิต
สาขาวิชาเทคโนโลยีอิเล็กทรอนิกส์ ภาควิชาเทคนิคอุตสาหกรรม
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2543

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

b11215380

หัวข้อปริญญานิพนธ์	**เครื่องหยอดเหรียญ**
	COINS COUNTER
นักศึกษา	**นาย เฉลิมชัย ลาปัน เลขประจำตัว 41013284**
	นาย ณัฐกุล บุญญะอำไพวงศ์ เลขประจำตัว 41013287
อาจารย์ที่ปรึกษา	**ผู้ช่วยศาตราจารย์ประดิษฐ์ วัชรพิบูลย์**
ภาควิชา	**เทคนิคอุตสาหกรรม**
ปีการศึกษา	2543

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบังอนุมัติให้นับปริญญานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาหลักสูตรอุตสาหกรรมศาสตรบัณฑิต

คณะกรรมการสอบปริญญานิพนธ์

_____ประธานกรรมการ
(_____)

_____กรรมการ
(_____)

_____กรรมการ
(_____)

_____กรรมการ
(_____)

_____กรรมการ
(_____)

_____กรรมการ
(_____)

ฉิขสิทธิ์คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญานิพนธ์	**เครื่องหยอดเหรียญ**
	COINS COUNTER
นักศึกษา	**นาย เฉลิมชัย ลาปัน เลขประจำตัว 41013284**
	นาย ณัฐกุล บุญญะอำไพวงศ์ เลขประจำตัว 41013287
อาจารย์ที่ปรึกษา	**ผู้ช่วยศาตราจารย์ประดิษฐ์ วัชรพิบูลย์**
ภาควิชา	**เทคนิคอุตสาหกรรม**
ปีการศึกษา	2543

บทคัดย่อ

ปริญญานิพนธ์นี้ นำเสนอเครื่องหยอดเหรียญตั้งเวลาได้ โดยใช้ไมโครคอนโทรลเลอร์ควบคุมหน้าที่การทำงานของกลไก จุดมุ่งหมายหลัก เพื่อเป็นการศึกษาถึงการทำงานของไมโครคอนโทรเลอร์ และ การทำงานของระบบกลไก และเพื่อเป็นการพัฒนาทักษะในการเขียนโปรแกรมใช้งานในระบบควบคุม ได้อย่างมีประสิทธิภาพสูงสุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Thesis Title Coins Counter

Student Mr. Chaluemchai Lapun ID 41013284

Mr. Nuttagul Boonyaumpaiwong ID 41013287

Advisor Assistant prof. Pradit Wacharapiboon

Academic Year 2000

ABSTRACT

This thesis presents the using of microcontroller to control the mechanical of the vending machine with programmable timer. The main purpose is to study the function of microcontroller BASIC STAMP 2 to development the skill of writing the PBASIC language.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้ สำเร็จลงด้วยความเรียบร้อยโดยได้รับการสนับสนุนและการให้คำปรึกษาจากหลายฝ่าย ฉะนั้นผู้จัดทำจึงขอขอบพระคุณท่านอาจารย์ประดิษฐ์ วัชรพิบูลย์ เป็นอย่างสูงในการแนะนำให้คำปรึกษาชี้แนะ และติชมแก้ไข ซึ่งเป็นอาจารย์ที่ปรึกษา คณะผู้จัดทำรู้สึกซาบซึ้งในความอนุเคราะห์จากท่าน

พร้อมทั้งขอขอบคุณ คุณสรพงษ์ แซ่เตีย ที่ให้ความช่วยเหลือและคำแนะนำทางด้านการออกแบบวงจรโดยใช้ Protel และ Orcad และขอบคุณเพื่อนๆ ทุกท่าน ที่ให้ความสนับสนุน คำชี้แนะ ซึ่งทำให้โครงงานนี้สำเร็จลุล่วงไปได้ด้วยดี ตามวัตถุประสงค์

คุณค่าและประโยชน์อันพึงมีจากคู่มือฉบับนี้ ผู้จัดทำขอมอบแด่ผู้มีพระคุณทุกท่าน

คณะผู้จัดทำ
นายเฉลิมชัย ลาปัน
นายณัฐกุล บุญญะอำไพวงศ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทคัดย่อภาษาไทย	ก
บทคัดย่อภาษาอังกฤษ	ข
กิตติกรรมประกาศ	ค
สารบัญ	ง
สารบัญตาราง	จ
สารบัญรูปภาพ	ฉ
บทที่ 1 บทนำ	
1.1 วัตถุประสงค์	1
1.2 แนวความคิดและที่มา	1
1.3 ประโยชน์ที่คาดว่าจะได้รับ	1
บทที่ 2 หลักการและการทำงานทั่วไป	
2.1 ภาคคอนโทรล	3
2.2 การทำงานของเครื่อง	19
บทที่ 3 การออกแบบ	
3.1 ภาคไมโครคอนโทรลเลอร์	23
3.2 ภาคตรวจนับจำนวนเหรียญ	24
3.3 ภาคควบคุมอุปกรณ์ระบบกลไก และ ภาคทอนเหรียญ	25
3.4 ภาคจ่ายไฟ	25
3.5 ภาคแสดงผล	25
3.6 ภาคเอาท์พุท	25
บทที่ 4 การพัฒนาการเขียนโปรแกรม	
4.1 โปรแกรมการใช้งาน	26
4.2 ชุดไมโครคอนโทรลเลอร์	27
4.3 ชุดการตรวจจับเหรียญ	27
บทที่ 5 การทดลองและวิธีการใช้งาน	
5.1 การทำงานในโหมดปรกติ	28
5.2 การทำงานในโหมดตั้งเวลา	30

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

ภาคผนวก

- ภาคผนวก ก. รายละเอียดทางฮาร์ดแวร์
- ภาคผนวก ข. โปรแกรมการใช้งาน
- ภาคผนวก ค. DATA SHEET

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่	หน้า
2.1 การจัดขาของ BS-IC	8
2.2 ตำแหน่งขาสัญญาณของพอร์ตอนุกรม	13
2.3 หน้าที่ของขาสัญญาณแต่ละเส้น	13
2.4 ตารางเปรียบเทียบมาตรฐานการสื่อสารข้อมูลของ EIA	17

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่	หน้า
2.1 ไดอะแกรมเบื้องต้นของเบสิกแสตมป์ 2	4
2.2 วงจรสมบูรณ์ของเบสิกแสตมป์ 2	6
2.3 การเชื่อมต่อเบสิกแสตมป์ 2 กับพอร์ตอนุกรมของคอมพิวเตอร์	6
2.4 การจัดขาของ BS2-IC	8
2.5 แสดงการจัดหน่วยความจำภายในของ BS2-IC	10
2.6 คอนเน็คเตอร์ แบบ DB-25 และคอนเน็คเตอร์แบบ DB-9	12
2.7 แสดงโครงสร้างลักษณะคุณสมบัติทางไฟฟ้าแบบไม่สมดุล	14
2.8 แสดงโครงสร้างลักษณะคุณสมบัติทางไฟฟ้าแบบสมดุล	15
2.9 แสดงโครงสร้างการสื่อสารข้อมูลแบบอนุกรมตามมาตรฐาน RS-485	16
2.10 ก. Flowchart การทำงานของเครื่อง	20
2.10 ข. Flowchart การทำงานของเครื่อง	21
3.1 บล็อกไดอะแกรมรวม	22
3.2 บล็อกไดอะแกรมรวมภาคไมโครคอนโทรลเลอร์	23
3.3 ลักษณะภายนอกของ BS2-IC	23

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

วัตถุประสงค์

1. เพื่อศึกษาการเขียนโปรแกรมที่ใช้ในงานควบคุม
2. เพื่อศึกษาการทำงานของไมโครคอนโทรลเลอร์
3. เพื่อศึกษาระบบกลไก ควบคุมการหยอดเหรียญขั้นพื้นฐาน
4. เพื่อศึกษาการเชื่อมต่อข้อมูลของระบบไมโครคอนโทรลเลอร์
5. เพื่อศึกษาระบบการทำงานแบบแมคคาโทรนิกส์
6. เพื่อศึกษาการนำไมโครคอนโทรลเลอร์ไปประยุกต์ใช้งานได้

แนวความคิดและที่มา

เนื่องจากในชีวิตประจำวัน การดำรงชีวิตของมนุษย์ในเชิงพาณิชย์ได้มีการกำหนดเวลาเพื่อกำหนดราคาของการใช้บริการสินค้านั้น อาทิเช่น INTERNET SERVICE เป็นร้านให้บริการอินเตอร์เนต โดยมีเงื่อนไข เช่น ให้บริการอินเตอร์เนตในระยะเวลา 60 นาที ต่อ 60 บาท หรือ 1 นาที ต่อ 1 บาท หรืออาจจะเป็นร้านเกมส์ ซึ่งให้บริการเกมส์ซึ่งต้องกำหนดระยะเวลา เช่น 60 นาที ต่อ 20 บาท ดังนั้นจึงเป็นแนวความคิด และที่มาที่จะทำเครื่องตั้งเวลาหยอดเหรียญ เพื่ออำนวยความสะดวกสบายให้กับผู้ประกอบการร้านค้าที่ต้องกำหนดเวลาเพื่อกำหนดราคาแก่ลูกค้า ทั้งยังสามารถนำไปประยุกต์ใช้ในชีวิตประจำวันอื่น ๆ ได้อีกมากมาย

ประโยชน์ที่คาดว่าจะได้รับ

1. ได้รับความรู้ความเข้าใจในการเขียนโปรแกรมควบคุมระบบไมโครคอนโทรลเลอร์
2. ได้รับความรู้ความเข้าใจในระบบกลไก ระบบควบคุมการหยอดเหรียญ
3. สามารถเข้าใจการเชื่อมต่อข้อมูลของไมโครคอนโทรลเลอร์
4. สามารถได้รับความรู้ความเข้าใจในระบบแมคคาโทรนิกส์
5. สามารถนำไปประยุกต์ใช้ในชีวิตประจำวันได้เป็นอย่างดี

ในการสร้างเครื่องหยอดเหรียญขึ้นมานี้ เพื่อศึกษาการเขียนและการพัฒนาโปรแกรม และการใช้ไมโครคอนโทรลเลอร์ในการควบคุมต่าง ๆ และนำมาควบคุมระบบแมคคานิค พร้อมทั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ศึกษาการเชื่อมต่อกับระบบแมคคานิค โดยมีไอซีไมโครคอนโทรลเลอร์ตระกูลของเบสิกแสตมป์ รุ่น BS2-IC ของบริษัท Parallax Inc.

ซึ่งเบสิกแสตมป์นี้จะใช้งบประมาณเริ่มต้นต่ำ เมื่อเทียบกับไมโครคอนโทรลเลอร์ตระกูลอื่น ๆ เนื่องจากเบสิกแสตมป์ไม่ต้องใช้เครื่องโปรแกรมใด ๆ ในการพัฒนาไม่ต้องใช้เครื่องเลียนแบบการทำงาน (emulator) หรือซอฟต์แวร์จำลองการทำงาน (simulator) ในการพัฒนาทางด้านการเขียนโปรแกรมด้วยชุดคำสั่งภาษาเบสิก ที่เรียกว่า พีเบสิก (PBASIC) ซึ่งมีด้วยกัน 36 คำสั่ง มีจำนวนอินพุท เอาท์พุท 16 ขา ความเร็วในการกระทำคำสั่ง 4,000 คำสั่งต่อวินาที ขนาดของหน่วยความจำโปรแกรม 2 กิโลไบต์ ความสามารถในการเก็บคำสั่งภาษาเบสิก 500 คำสั่ง จำนวนรีจิสเตอร์ 26 ไบต์ อัตราเร็วในการถ่ายทอดข้อมูล 5 กิโลไบต์ต่อวินาที การเก็บรักษาข้อมูล 10 ปี โปรแกรมใหม่ได้ 10 ล้านรอบ และการทำงานทั้งหมดของระบบใช้ แอลซีดี (LCD) แบบโมดูล ขนาด 16 ตัวอักษร เป็นตัวแสดงผลข้อมูลแสดงจำนวนเวลาเหรียญใช้ LED ตัวเลข 7 ส่วน แบบ 2 หลัก และ ส่วนแสดงเวลาใช้แบบ 4 หลัก เป็นแบบแคโทดร่วม และใช้โซลินอยด์ (solinoids) ควบคุมการทอนเหรียญ

เครื่องหยอดเหรียญนี้มีการติดต่อกับผู้ใช้ประกอบด้วยฟังก์ชั่น (Function)

- ตั้งเวลาหยอดเหรียญ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
หลักการทำงานทั่วไป

จากหลักการทำงานโดยทั่วไปจะมีการทำงานที่เราสามารถแบ่งเป็นภาคใหญ่ ๆ ได้ คือ ภาคคอนโทรล (control) โดยส่วนแสดงผลจะใช้ แอลซีดี (LCD) และLED ตัวเลข 7 ส่วนในการแสดงผล ซึ่งการแสดงผลจะมีการแสดงผลจำนวนเงินหรือจำนวนเหรียญที่หยอดเข้าไป และจะมีส่วนที่แสดงเวลา และยังมีฟังก์ชั่น (Function) การโปรแกรมการทำงานของเครื่องคือจำนวนเวลาต่อเหรียญ 1 เหรียญโดยจะมีคีย์สำหรับการโปรแกรมได้โดยตัวผู้ใช้เอง โดยการทำงาน เราสามารถแบ่งได้เป็นส่วน ๆ ดังนี้

2.1 ภาคคอนโทรล

ประกอบไปด้วยไอซีหลาย ๆ ตัวมาทำงานร่วมกันประกอบไปด้วย

2.1.1 IC เบสิกแสตมป์ รุ่น BS2-IC ของบริษัท Parallax Inc.

ไมโครคอนโทรเลอร์ชิพเดี่ยวตระกูลของเบสิกแสตมป์นี้ผลิตโดยบริษัท Parallax Inc. มีอยู่ด้วยกัน 4 รุ่น คือ

1. BS1-REV.D หรือ เบสิกแสตมป์ 1 รีวิชั่นดี มีขาพอร์ตอินพุท เอาท์พุท 8 ขา ขนาดหน่วยความจำ 256 ไบต์ บรรจุคำสั่งภาษา PBASIC-1 ได้ 75 คำสั่ง ความเร็วในการกระทำคำสั่งภาษาเบสิก 2,000 คำสั่งต่อวินาที

2. BS1-IC หรือ เบสิกแสตมป์ 1 รุ่นไอซี มีรูปร่างเป็นไอซีแบบแถวเดี่ยว มีจำนวนขาอินพุทเอาท์พุท, ขนาดหน่วยความจำ และความเร็วเท่ากับรุ่น REV.D

3. BS2-IC หรือ เบสิกแสตมป์ 2 รุ่น ไอซี มีรูปร่างและขนาดเท่ากับไอซีตัวถัง DIP ขนาด 24 ขา มีขาพอร์ตอินพุทเอาท์พุท 16 ขา ขาสำหรับเชื่อมต่อกับคอมพิวเตอร์ 3 ขา ขนาดหน่วยความจำ 2 กิโลไบต์ บรรจุคำสั่งภาษา PBASIC-2 ได้ 500 คำสั่ง มีความเร็วในการประมวลผลคำสั่งภาษา เบสิก 4,000 คำสั่งต่อวินาที

4. BS2 SX-IC หรือ เบสิกแสตมป์ 2 รุ่นเอสเอ็กซ์ เป็นรุ่นที่ใช้ ไมโครคอนโทรลเลอร์ตระกูล SX ซึ่งมีความเร็วในการทำงานสูงกว่า BS2-IC มีจำนวนขาพอร์ตอินพุทเอาท์พุทและขนาดของ ตัวถังเหมือนกับ BS2-IC ขนาดของหน่วยความจำสูงถึง 16 กิโลไบต์ สามารถบรรจุคำสั่งภาษา PBASIC-2 ได้ 2,000 คำสั่ง มีความเร็วในการประมวลผลคำสั่งภาษาเบสิก 1,000 คำสั่งต่อวินาที เหมาะสำหรับงานที่ต้องการความเร็วและขนาดของหน่วยความจำสูง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

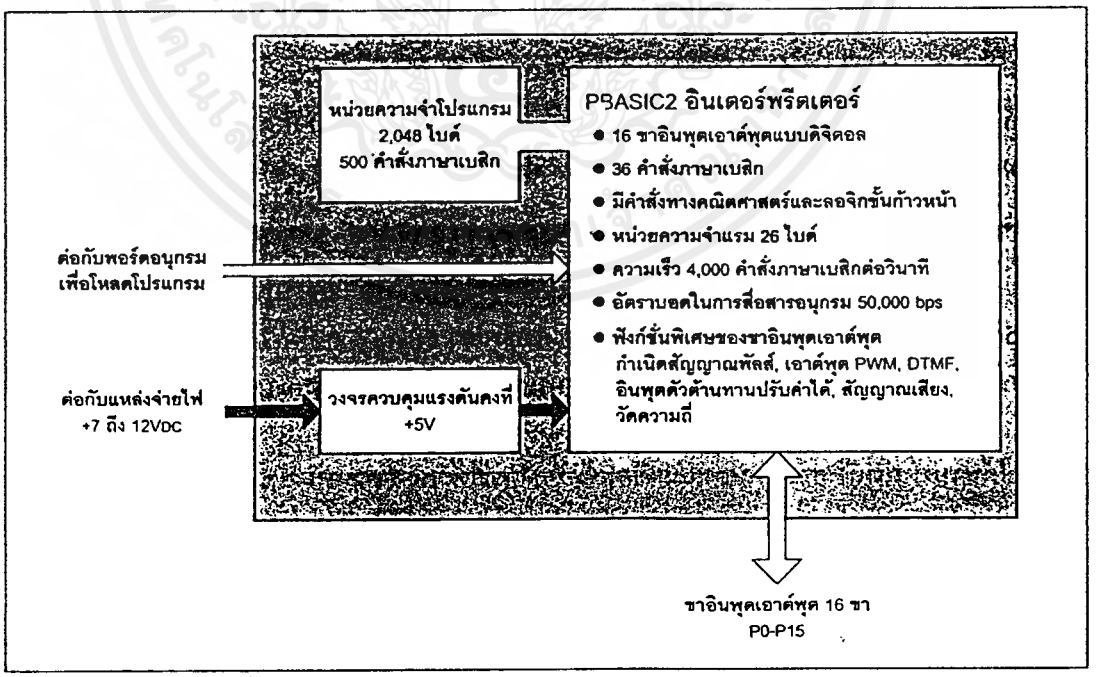
ตระกูล (BASIC Stamp) ได้ถูกออกแบบมาด้วยแนวคิดที่ว่า ผู้ใช้งานไม่จำเป็นต้องมีความรู้เกี่ยวกับภาษาแอสแซมบีของไมโครคอนโทรลเลอร์เบอร์ใด ๆ เพียงสามารถเขียนโปรแกรมด้วยภาษาเบสิก อันเป็นภาษาที่มีความซับซ้อนน้อยที่สุดก็สามารถเขียนโปรแกรมเพื่อควบคุมและใช้งานไมโครคอนโทรลเลอร์ได้

ในขณะเดียวกันต้องไม่ใช้เครื่องมือในการพัฒนาที่มีความซับซ้อน กล่าวคือ เพียงต่อสายจากพอร์ตของคอมพิวเตอร์ ส่งข้อมูลผ่านโปรแกรมสื่อสารลงสู่หน่วยความจำบนตัวบอร์ด ปลดสายออกทำการรีเซตระบบ เพียงเท่านี้บอร์ดไมโครคอนโทรลเลอร์นั้นก็สามารถทำงานได้แล้ว

แนวความคิดนี้ได้รับการต้อนรับเป็นอย่างดีจากนักทดลองไมโครคอนโทรลเลอร์โดยเฉพาะอย่างยิ่งนักพัฒนาระบบควบคุมของหุ่นยนต์เนื่องจากขนาดของบอร์ดเล็กมากกินไฟฟ้าต่ำเขียนโปรแกรมง่าย ไม่ต้องใช้เครื่องมือพัฒนาราคาแพง

โครงสร้างและคุณสมบัติของ BS2-IC

ในรูปที่ 2.1 เป็นไดอะแกรมการทำงานเบื้องต้นพร้อมรายละเอียดสำคัญทางฮาร์ดแวร์ของเบสิกแสตมป์ 2 หรือ BS2 ส่วนกระดอบหลักมี.3 ส่วน คือ ตัวแปลงภาษา PBASIC-2 , หน่วยความจำโปรแกรม และวงจรเรกูเลเตอร์



รูปที่ 2.1 ไดอะแกรมเบื้องต้นของเบสิกแสตมป์ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วงจรสมบูรณ์ของ BS2 แสดงในรูปที่ 2.2 แบ่งออกเป็น 4 ส่วนเริ่มจาก

- **IC1**

หัวใจหลักของเบสิกแสตมป์อยู่ที่ IC1 เป็นไมโครคอนโทรลเลอร์เบอร์ PIC16C57 ซึ่งภายในบรรจุโปรแกรมแปลงภาษาเบสิก หรือ PBASIC-2 อินเตอร์พรีตเตอร์เอาไว้ ไม่สามารถเข้ามาทำการแก้ไขได้ IC1 ทำงานด้วยสัญญาณนาฬิกาความถี่ 20 MHz

BS2 มีจำนวนขาอินพุท เอาท์พุท 16 เส้น คือขาP0-P15 ซึ่งต่อตรงเข้ากับขาอินพุท เอาท์พุท ของ IC1 มีความสามารถในการจ่ายกระแสของเอาท์พุท ในลักษณะกระแสซิงก์ 25mA ต่อขาและกระแสซอร์ส 20 mA ต่อขา ส่วนระดับแรงดันจะเป็นไปตามมาตรฐานทีทีแอล (TTL) คือมีขนาด+5V ที่ลอจิก " 1 " และ 0V ที่ลอจิก " 0 " ดังนั้นจึงสามารถใช้ขาเอาท์พุทของ BS2 ขับ LED ได้โดยไม่ต้องอาศัยอุปกรณ์ขับกระแสหรือไดรเวอร์ (driver) เพิ่มเติม

- **IC2**

ทางด้านหน่วยความจำโปรแกรม IC2 ใช้ไอซีหน่วยความจำอีอีพรอมอนุกรมเบอร์ 24LC16 มีความจุ 2 กิโลไบต์ (2KB) สามารถบรรจุคำสั่งของ PBASIC-2 ที่ใช้ในการรันโปรแกรมของ BS2 ได้ 500 คำสั่ง การเก็บรักษาข้อมูลทำได้ยาวนานและมีรอบของโปรแกรม คือ เก็บได้นาน 10 ปี โปรแกรมใหม่ได้ 10 ล้านรอบ

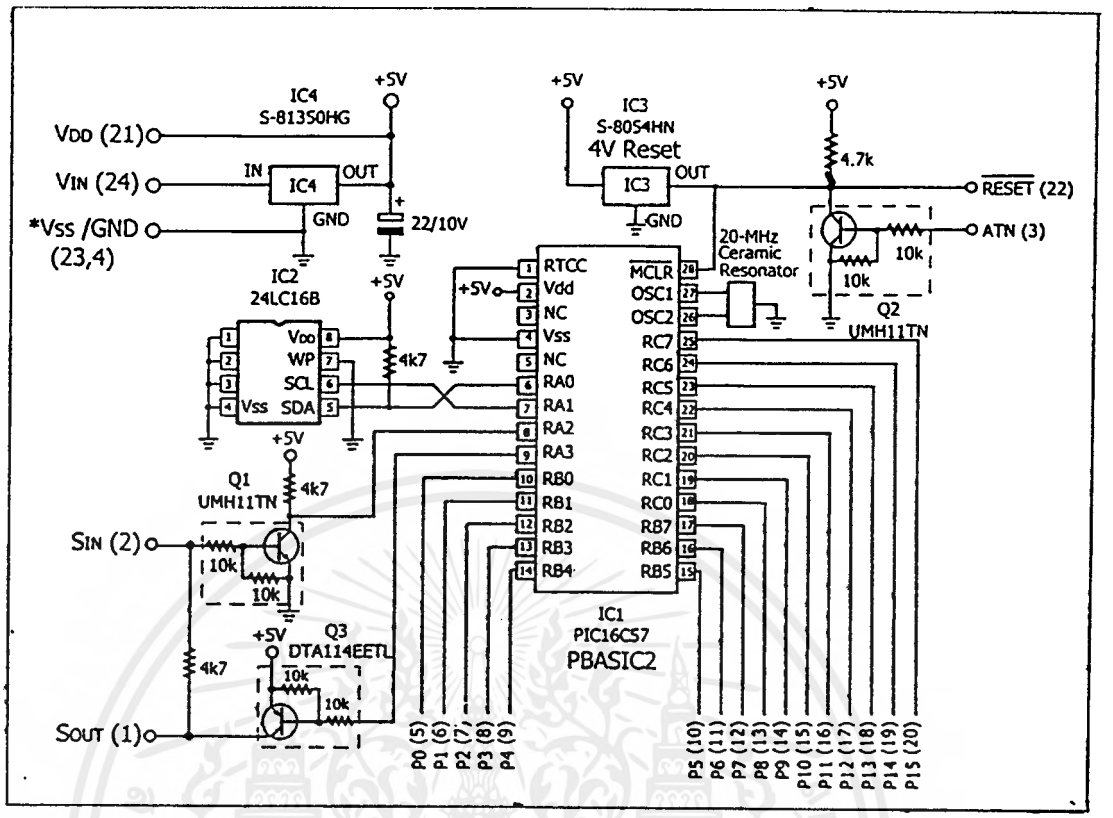
- **IC3**

วงจรรีเซตของ BS2 ใช้ IC3 เป็นตัวทำหน้าที่นี้ เมื่อเริ่มจ่ายไฟให้แก่ BS2 IC3 จะเริ่มทำงานแล้วสร้างสัญญาณรีเซตให้แก่ IC1 โดยป้อนสัญญาณรีเซตเข้าที่ขา MCLR ของ IC1 ในกรณีที่แรงดันไฟเลี้ยง +5V เกิดลดต่ำลงกว่า +4 V ไม่ว่าจะด้วยสาเหตุของแบตเตอรี่ที่ใช้เป็นแหล่งจ่ายไฟมีระดับแรงดันลดลง หรือวงจรแหล่งจ่ายไฟภายนอกของ BS2 เกิดลัดวงจร ทำให้แรงดันต่ำลง IC3 ก็จะทำงานสร้างสัญญาณรีเซต ให้แก่ IC1 เพื่อควบคุมให้ IC1 อยู่ในสภาวะรีเซตจนกว่าแรงดันไฟเลี้ยงของ BS2 จะกลับมาอยู่ในสภาวะปกติ

- **IC4**

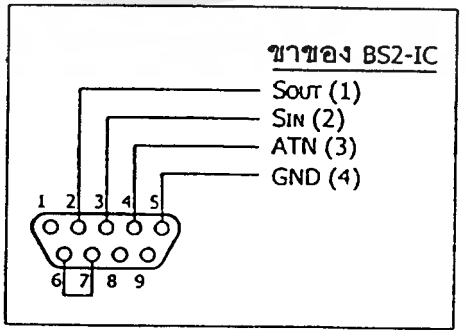
BS2 สามารถรับไฟเลี้ยงได้ทั้ง +5V และสูงกว่าโดยแรงดันไฟเลี้ยงที่สูงกว่า +5V คือ ตั้งแต่ 6-13V ให้ป้อนเข้าที่ขา Vin ภายใน BS2 จะมี IC4 เป็นไอซีเรกูเลเตอร์ ควบคุมให้เกิดแรงดันคงที่ที่+5V เพื่อจ่ายให้แก่วงจรทั้งหมดของ BS2 ในกรณีป้อนแรงดัน +5V ให้แก่ BS2 ต้องต่อเข้าที่ขา VDD

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.2 วงจรสมบูรณ์ของเบสิกแสตมป์ 2

ส่วนการโปรแกรมข้อมูล ยังคงใช้การโปรแกรมในลักษณะอนุกรม แต่จะติดต่อกับพอร์ตอนุกรมคอมพิวเตอร์แทน โดยต่อสายจากขา TxD, RxD, DTR และกราวด์ของพอร์ตอนุกรมเข้าสู่ขา Sin, Sout, ATN และกราวด์ของ BS2 นอกจากนั้นที่ขา 6 และ 7 ของพอร์ตอนุกรมต้องต่อถึงกันด้วย เพื่อให้สามารถโปรแกรมข้อมูลบน BS2 ได้ตลอดเวลาอย่างอัตโนมัติ ดังแสดงในรูปที่ 2.3



รูปที่ 2.3 การเชื่อมต่อเบสิกแสตมป์ 2 กับ พอร์ตอนุกรมของคอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สัญญาณจากพอร์ตอนุกรมของคอมพิวเตอร์โดยทั่วไปมีระดับแรงดันสูงกว่าระดับทีทีแอล กล่าวคือ มีแรงดันประมาณ ±12V สำหรับคอมพิวเตอร์ตั้งโต๊ะ และ ±6V ถึง ±8V สำหรับคอมพิวเตอร์โน้ตบุ๊ก ดังนั้นจึงต้องมีวงจรแปลงระดับสัญญาณจากคอมพิวเตอร์ให้เป็นระดับทีทีแอล (0-5V) เพื่อให้สามารถติดต่อกับ PIC16C57 ได้

ขา SIN เป็นขาสำหรับรับข้อมูลจากคอมพิวเตอร์เมื่อขาคอมพิวเตอร์ส่งข้อมูล "1" มา จะทำให้ SIN เป็นลบ ทรานซิสเตอร์ Q1 จะไม่ทำงาน ทำให้ขาที่ RA2 ของ PIC16C57 (หรือ IC1) มีแรงดัน +5V เกิดเป็นลอจิก "1" ในทางตรงกันข้ามเมื่อคอมพิวเตอร์ส่งข้อมูล "0" ขา SIN จะเป็นบวก ทรานซิสเตอร์ Q1 ทำงานส่งผลให้ขา RA2 เสมือนต่อลงกราวด์ มีลอจิกเป็น "0"

ขา SOUT ใช้สำหรับส่งข้อมูลจาก BS2 ไปยังคอมพิวเตอร์ เมื่อ BS2 ส่งข้อมูล "1" ทรานซิสเตอร์ Q3 ไม่ทำงาน ที่ขา SOUT จะได้รับแรงดันลบจากขา SIN ผ่านทางตัวต้านทาน 4.7k ทำให้ที่ขา SOUT มีระดับแรงดันเป็นลบ ซึ่งคอมพิวเตอร์จะอ่านเป็นลอจิก "1" ทั้งนี้เนื่องจากในมาตรฐาน RS-232 แล้วลอจิก "1" คือระดับแรงดันตั้งแต่ –3 ถึง –12V ในขณะที่ลอจิก "0" คือระดับแรงดัน +3 ถึง +12V เมื่อ BS2 ส่งข้อมูล "0" ทรานซิสเตอร์ Q3 ทำงานที่ขา SOUT จึงเกิดแรงดัน +5V ทำให้คอมพิวเตอร์อ่านข้อมูลได้เป็น "0"

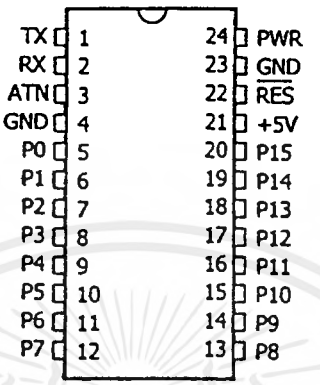
เมื่อเป็นเช่นนี้ในการติดต่อระหว่าง BS2 กับคอมพิวเตอร์จะต้องสลับกันรับและส่งข้อมูล กล่าวคือ เมื่อคอมพิวเตอร์ส่งข้อมูลมา BS2 ต้องทำหน้าที่รับข้อมูลอย่างเดียว ไม่สามารถที่จะส่งข้อมูลกลับไปยังคอมพิวเตอร์ในเวลาเดียวกันได้

ขา ATN ซึ่งต่อเข้ากับขา DTR (Data Terminal Ready) ใช้ในการแฮนด์เชก หรือตรวจสอบความพร้อมในการรับส่งข้อมูลของพอร์ตอนุกรม จะมีลักษณะทำงานคล้ายกับขา SIN ถ้าหากคอมพิวเตอร์ส่งข้อมูล "0" หรือทำให้ขา DTR มีแรงดันเป็น +12V ทรานซิสเตอร์ Q2 ทำงาน ทำให้ขาคอลเล็กเตอร์ของ Q2 เสมือนต่อลงกราวด์เท่ากับว่าขา MCLR ของ PIC16C57 ถูกต่อลงกราวด์ด้วยอันเป็นการสร้างสัญญาณรีเซตให้แก่ PIC16C57 ดังนั้นในขณะที่ทำการโปรแกรมข้อมูลลงบน BS2 ซอฟต์แวร์ที่ใช้ในการโปรแกรม ซึ่งเรียกว่า STAMP2 โฮสต์โปรแกรม จะส่งพัลส์มายังขา ATN เพื่อรีเซต IC1 แล้วตามด้วยการส่งข้อมูลมายังขา SIN เพื่อแจ้งให้ทราบว่าต้องการเขียนโปรแกรมใหม่ลงบน BS2 แต่ถ้าหาก STAMP2 โฮสต์โปรแกรม ทำให้ขา ATN เป็น –12V หรือเป็นลอจิก "1" ก็จะหมายความว่าขณะนี้ BS2 อยู่ในโหมดรัน นั่นคืออยู่ในโหมดทำงานปรกติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. การจัดขาลักษณะภายนอกของเบสิกแสตมป์ รุ่น BS2-IC

ในรูปที่ 2.4 แสดงการจัดขาตามลักษณะภายนอกของชิพ BS2-IC



รูปที่ 2.4 การจัดขาของ BS2-IC

ตารางที่ 2.1 การจัดขาของ BS2-IC

ขาที่	ชื่อขา	หน้าที่
1	Tx หรือ Sout	ส่งข้อมูลออกแบบอนุกรม เชื่อมต่อกับขา RxD (ขา 2) ของพอร์ตอนุกรมของคอมพิวเตอร์ ผ่านคอนเน็คเตอร์ DB-9
2	Rx หรือ Sin	รับข้อมูลเข้าแบบอนุกรม เชื่อมต่อกับขา TxD (ขา 3) ของพอร์ตอนุกรมของคอมพิวเตอร์ ผ่านคอนเน็คเตอร์ DB-9
3	ATN	ขารีเซตลอจิกสูง เชื่อมต่อกับขา DTR (ขา 4) ของพอร์ตอนุกรมของ คอมพิวเตอร์ผ่านคอนเน็คเตอร์ DB-9
4	Vss/GND	ขากราวด์ เชื่อมต่อกับขา GND (ขา 5) ของพอร์ตอนุกรมของคอมพิวเตอร์ ผ่านคอนเน็คเตอร์ DB-9

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.1 การจัดขาของ BS2-IC (ต่อ)

ขาที่	ชื่อขา	หน้าที่
5-20	P0-P15	ขาพอร์ตอินพุทเอาท์พุท 16 ขา
21	+5V	ขาไฟเลี้ยง +5V จากเรกูเลเตอร์ภายใน BS2-IC
22	RES	ขารีเซตลอจิกต่ำ สำหรับรีเซต BS2-IC
23	Vss/GND	ขากราวด์ของ BS2-IC
24	*PWR/Vin	ขาอินพุทสำหรับวงจรเรกูเลเตอร์ภายใน BS2-IC สามารถรับแรงดัน 7-12V

*ในกรณีจ่ายไฟเข้าที่ขา 24 (PWR) ไม่จำเป็นต้องจ่ายไฟหรือใช้แรงดันที่ขา 21

การจัดหน่วยความจำภายในเบสิกแสตมป์ 2

เบสิกแสตมป์ 2 มีหน่วยความจำอยู่ 2 ชนิดอยู่บนตัวมัน ชนิดแรกคือ หน่วยความจำแรม (RAM : Random Access Memory) ซึ่งใช้สำหรับกำหนดค่าตัวแปรในการเขียนโปรแกรม และชนิดที่ 2 คือ หน่วยความจำอีอีพรอม (EEPROM : Electrically Erasable Programmable Read-Only Memory) ใช้สำหรับการเก็บโปรแกรมและข้อมูลที่ต้องการคงอยู่เป็นระยะเวลายาวนาน เปรียบเทียบกับคอมพิวเตอร์แล้ว หน่วยความจำอีอีพรอมก็เหมือนกับฮาร์ดดิสก์นั่นเอง ซึ่งใช้ในการเก็บทั้งโปรแกรมและแฟ้มข้อมูล

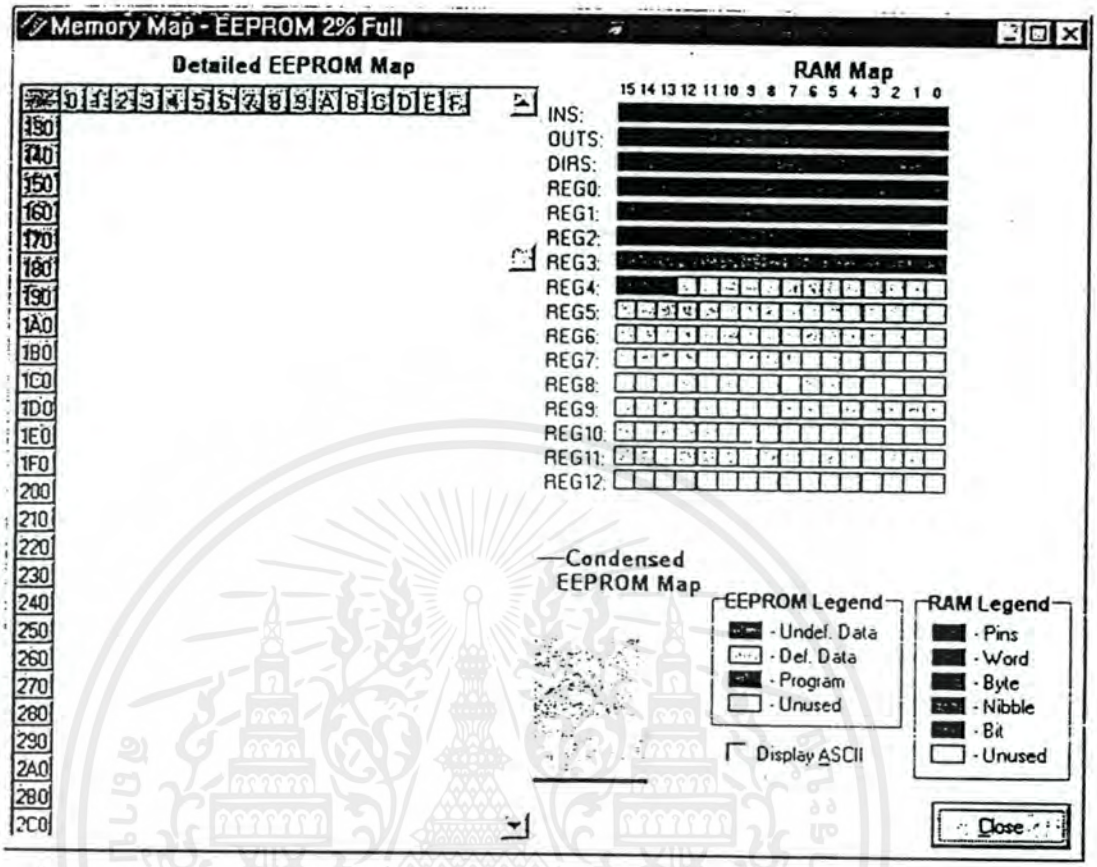
ข้อแตกต่างระหว่างหน่วยความจำแรมกับหน่วยความจำอีอีพรอมคือ

1. ข้อมูลในหน่วยความจำแรมจะสูญหายไปเมื่อไม่มีไฟจ่ายให้เบสิกแสตมป์ 2 และเมื่อจ่ายไฟให้กับเบสิกแสตมป์ 2 อีกครั้งค่าต่างๆ ภายในหน่วยความจำแรมจะกลายเป็น 0
2. หน่วยความจำอีอีพรอมจะเก็บรักษาข้อมูลเอาไว้แม้ไม่ได้จ่ายไฟเลี้ยง และข้อมูลภายในหน่วยความจำอีอีพรอมนี้จะคงอยู่ตลอดจนกว่าจะมีการเขียนข้อมูลใหม่หรือมีการโหลดโปรแกรมจากคอมพิวเตอร์ลงไปยังเบสิกแสตมป์ 2 ครั้งใหม่

หน่วยความจำข้อมูลของเบสิกแสตมป์ 2

เบสิกแสตมป์ 2 มีหน่วยความจำข้อมูล 32 ไบต์ โดย 6 ไบต์สงวนไว้สำหรับเก็บข้อมูลของอินพุทเอาท์พุท และควบคุมทิศทางของขาอินพุทเอาท์พุท ดังนั้นจึงเหลือหน่วยความจำ 26 ไบต์สำหรับการใช้งานทั่วไป รูปที่ 2.5 แสดงให้เห็นถึงการจัดการหน่วยความจำของเบสิกแสตมป์ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.5 การจัดหน่วยความจำภายในของ BS2-IC

ตัวแปรอินพุท/เอาท์พุท

จากรูปที่ 2.5 รีจิสเตอร์ 3 ตัวแรกจะใช้ในการติดต่อกับบิตอินพุทเอาท์พุท ทั้ง 16 บิต ซึ่งก็คือขาพอร์ต PO-P15 ของเบสิกแสตมป์ 2 นั่นเอง โดยรีจิสเตอร์ตัวแรกคือ INS ใช้สำหรับอ่านค่าอินพุทจากขา PO ถึง P15 โดยรีจิสเตอร์ INS จะสามารถอ่านค่าได้อย่างเดียว ไม่สามารถเขียนค่าไปยังตัวมันได้ ในขณะที่รีจิสเตอร์ OUTS จะใช้สำหรับส่งค่าออกไปยังขา PO-P15 เมื่อมีการกำหนดให้ขาพอร์ตนี้เป็นขาเอาท์พุท พร้อมทั้งสามารถแลตช์ค่าเอาไว้ได้ด้วย ส่วนรีจิสเตอร์ DIRS ใช้เพื่อควบคุมทิศทางของขาพอร์ต PO-P15 หรือใช้ในการกำหนดว่าขาใดเป็นอินพุทหรือเอาท์พุท วิธีการกำหนดหน้าที่ของขาพอร์ตทำได้ง่ายๆ ด้วยการเขียนข้อมูล " 0 " เมื่อต้องการให้ขาพอร์ตใดเป็นอินพุท และเขียนข้อมูล " 1 " เมื่อต้องการให้ขาพอร์ตนั้นๆ เป็นเอาท์พุท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อจ่ายไฟให้กับเบสิกแสตมป์ 2 ครั้งแรก ค่าของหน่วยความจำทุกไบต์จะเคลียร์เป็น 0 ไม่เว้นแม้แต่รีจิสเตอร์ DIRS (DIRS = %0000000000000000) ดังนั้นเมื่อจ่ายไฟครั้งแรก ขาพอร์ตทั้งหมด จึงเป็นอินพุท เมื่อต้องการให้ขาพอร์ตใดเป็นเอาท์พุทให้เขียนข้อมูล " 1 " ไปยังรีจิสเตอร์ DIRS ในตำแหน่งของแต่ละขาพอร์ตนั้น เมื่อกำหนดให้เป็นขาเอาท์พุทแล้ว จะสามารถส่งค่าออกไปได้ด้วยการเขียนข้อมูลไปยังรีจิสเตอร์ OUTS ถ้าหากต้องการเก็บค่าที่ขาเอาท์พุทนั้นก็สามารถทำได้ โดยการอ่านค่าจากรีจิสเตอร์ด้วยคำสั่ง INS

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การติดต่อสื่อสารผ่านพอร์ตอนุกรม

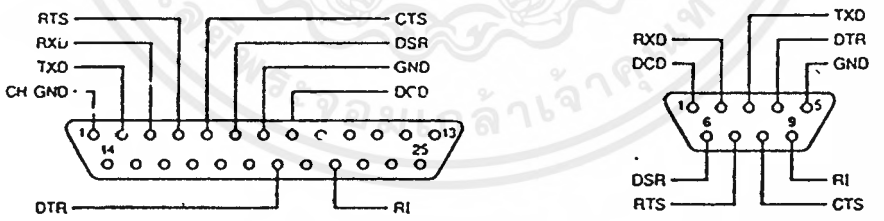
ลักษณะทางฮาร์ดแวร์

อุปกรณ์ที่ใช้การสื่อสารแบบอนุกรมสามารถแยกออกได้เป็น 2 ประเภทคือ DCE (Data Communication Equipment) อุปกรณ์เหล่านี้ได้แก่ โมเด็ม, TA และอะแดปเตอร์, พล็อตเตอร์ ฯลฯ และ DTE (Data Terminal Equipment) ซึ่งก็คือคอมพิวเตอร์นั่นเอง

ข้อกำหนดทางไฟฟ้าของพอร์ตอนุกรมได้ถูกกำหนดเป็นมาตรฐานโดย EIA (Electronics Industry Association) หรือ RS-232 ซึ่งประกอบไปด้วยสิ่งต่าง ๆ เหล่านี้

1. ช่วงไม่มีข้อมูล (space) หรือลอจิก "0" ต้องมีแรงดันอยู่ในช่วง –3 และ –25V
2. ช่วงข้อมูล (mark) หรือลอจิก "1" ต้องมีแรงดันอยู่ในช่วง +3 และ +25V
3. แรงดันในช่วง –3V ถึง +3V ไม่มีการนิยามไว้
4. แรงดันในขณะเปิดวงจรต้องมีค่าไม่เกิน 25V
5. กระแสขณะช็อตวงจรมีค่าไม่เกิน 500 mA

พอร์ตอนุกรมนี้จะมีอยู่ด้วยกัน 2 ขนาดคือคอนเน็คเตอร์แบบ D-type ตัวผู้ขนาด 25 Pin รูปที่ 2.6 (ก) และคอนเน็คเตอร์แบบ D-type ตัวผู้เช่นกันขนาด 9 Pin รูปที่ 2.6(ข) ซึ่งคอนเน็คเตอร์ทั้ง 2 แบบนี้จะติดอยู่ที่ด้านหลังของเครื่องคอมพิวเตอร์ ตารางที่ 2.2 แสดงตำแหน่งขาสัญญาณของพอร์ตอนุกรม



รูปที่ 2.6 (ก) คอนเน็คเตอร์แบบ DB-25 (ข) คอนเน็คเตอร์แบบ DB-9

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.2 ตำแหน่งขาสัญญาณของพอร์ตอนุกรม

D-type 25 Pin No.	D-type 9 Pin No.	Abbreviation	Full Name
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

ตารางที่ 2.3 หน้าที่ของขาสัญญาณแต่ละเส้น

Abbreviation	Full Name	Function
TD	Transmit Data	- Serial Data Output (TXD)
RD	Receive Data	- Serial Data Input (RXD)
CTS	Clear to Send	- This line indicates that the Modem is ready to exchange data.
DCD	Data Carrier Detect	- When the modem detects a "Carrier" from the modem at the other end of the phone line, this line becomes active.
DSR	Data Set Ready	- This tells the UART that the modem is ready to establish a link.
DTR	Data Terminal Ready	- This is the opposite to DSR. This tells the Modem that the UART is ready to link.
RTS	Request to Send	- This line informs the Modem that the UART is ready to exchange data.
RI	Ring Indicator	- Goes active when modem detects a ringing signal from the PSTN.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การสื่อสารข้อมูลแบบอนุกรมตามมาตรฐาน RS-232C

การสื่อสารข้อมูลแบบอนุกรมที่ใช้งานอยู่ในปัจจุบันนี้ได้มีการกำหนดมาตรฐานการรับส่งข้อมูลไว้หลายแบบด้วยกัน แต่ที่ได้รับความนิยมนำไปใช้กันอย่างแพร่หลาย คือการสื่อสารข้อมูลแบบอนุกรมตามมาตรฐาน RS-232C เหตุที่เป็นที่นิยมเนื่องจากเป็นการสื่อสารข้อมูลที่ใช้ในเครื่องคอมพิวเตอร์ (Computer) IBM PC ซึ่งเป็นเครื่องคอมพิวเตอร์ที่ใช้อย่างแพร่หลายจากอดีตมาถึงปัจจุบันมาตรฐาน RS-232C

โครงสร้างการสื่อสารข้อมูลเป็นแบบจุดต่อจุดเท่านั้นโดยมีลักษณะสมบัติทางไฟฟ้าและทางกายภาพ แสดงดังรูป

รูป 2.7 โครงสร้างลักษณะคุณสมบัติทางไฟฟ้าแบบไม่สมดุล

ข้อจำกัดของการสื่อสารข้อมูลแบบอนุกรมมาตรฐาน RS-232C

ข้อจำกัดนี้ประกอบกันด้วย 3 อย่างด้วยกันคือ

- ระยะทางในเคเบิล (Cable) RS-232C ค่าความจุไฟฟ้าระหว่างสายสัญญาณต่าง ๆ กับกราวด์เป็นพารามิเตอร์หลักที่จำกัดระยะทางการใช้เคเบิลนั้น ตามข้อกำหนดมีว่าค่าความจุไฟฟ้าที่มองไปจากวงจรขับ (Driver) จะต้องไม่เกิน 2500pF การเพิ่มระยะทางของเคเบิล ก็จะเพิ่มค่าความจุไฟฟ้าที่มาเป็นภาระของวงจรขับ ธรรมดาที่ใช้เคเบิลที่ใช้อยู่จะมีความจุไฟฟ้า 2500pF ที่ความยาวประมาณ 15 เมตร
- อัตราการส่งข้อมูล จะต้องไม่เกิน 20,000 บิต/วินาที ปัญหานี้เกี่ยวข้องกับความต้านทานกับค่าความจุของสายเคเบิล (Cable) เช่นกัน RS-232C นั้นกำหนดค่า ความต้านทานอินพุท (Input) ของวงจรภาครับค่อนข้างสูง 3000-7000Ω
- สัญญาณรบกวน สามารถทำให้เกิดข้อมูลผิดพลาด ซึ่งเกิดจากแหล่งกำเนิดภายนอก เช่น มอเตอร์ไฟฟ้า สถานีวิทยุ และจากแหล่งกำเนิดภายใน เช่นความต้านทานไม่สมบูรณ์ การกระจายของกระแสจากสายสัญญาณต่าง ๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การสื่อสารข้อมูลแบบอนุกรมตามมาตรฐาน RS-422A

ในการออกแบบการสื่อสารข้อมูลที่กล่าวมา ได้มีการพยายามที่จะออกแบบให้การสื่อสารข้อมูลได้รวดเร็วยิ่งขึ้น และมีระยะทางในการสื่อสารข้อมูลได้มากขึ้นด้วย ซึ่งที่ผ่านมาการสื่อสารข้อมูล มาตรฐาน RS-232C ได้ออกแบบมาเพื่อใช้เชื่อมโยงกับโมเด็ม (Modem) เท่านั้นจึงไม่ได้คำนึงถึงความเร็วและระยะทางในการสื่อสาร แต่ในปัจจุบันได้มีการออกแบบมาเพื่อรองรับความต้องการของการใช้งานที่ต้องการให้รับ-ส่งข้อมูลได้ไกลยิ่งขึ้น คือมาตรฐาน RS-422A ซึ่งจะใช้สัญญาณแบบดิฟเฟอร์เรนเชียล (Differential) ดังในรูป

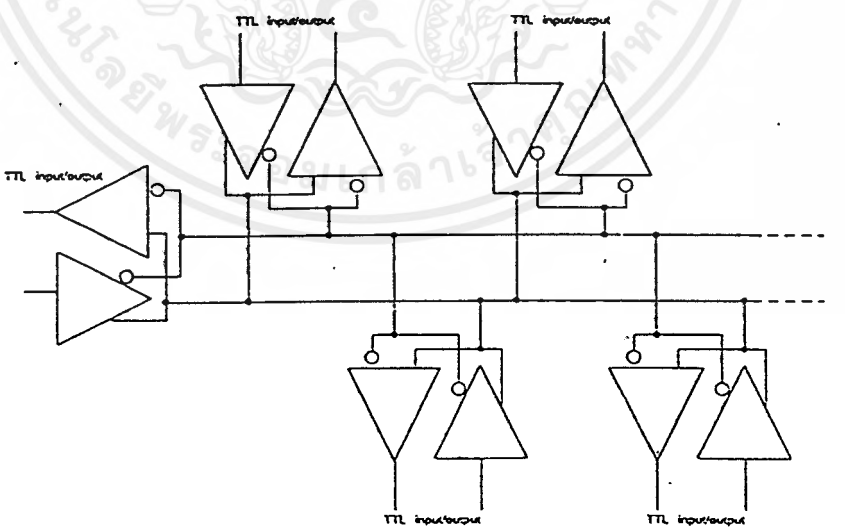
รูปที่ 2.8 โครงสร้างลักษณะคุณสมบัติทางไฟฟ้าแบบสมดุล

หลักการก็คือสัญญาณที่จะรับ-ส่งจะเป็นการเปรียบเทียบระหว่างสัญญาณ 2 เส้นเปรียบเทียบกับมาตรฐาน RS-232C ที่สัญญาณทุกสัญญาณจะเทียบกราวด์ (Ground) ซึ่งในการสื่อสารในระยะทางไกล ๆ แล้วสัญญาณจะถูกลดทอนไปถึงจุด ๆ หนึ่งสัญญาณนั้นก็จะผิดพลาดไปจากความเป็นจริงก็จะทำการรับ-ส่งผิดพลาดขึ้น แต่สำหรับสัญญาณดิฟเฟอร์เรนเชียล (Differential) แล้วการลดทอนของสัญญาณจะไปลดทอนทั้งสองสายด้วยค่าที่เท่ากันหรือใกล้เคียงกันและความแตกต่างของสัญญาณทั้งสองเส้น จากตัวส่งไปยังตัวรับก็ยังคงมีค่าเท่าเดิมหรือแตกต่างกันเล็กน้อย จึงทำให้ผลการลดลงของสัญญาณที่ระยะทางการสื่อสารไกลมีผลต่อสัญญาณดิฟเฟอร์เรนเชียล (Differential) มีค่าน้อยกว่า การสื่อสารข้อมูลแบบนี้จึงสามารถส่งข้อมูลได้ไกลกว่าและอัตราการสื่อสารข้อมูลสูงกว่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การสื่อสารข้อมูลแบบอนุกรมตามมาตรฐาน RS-485

การสื่อสารข้อมูลตามมาตรฐานที่กล่าวมาข้างต้น นั้นเป็นมาตรฐานการสื่อสารข้อมูลในแบบที่ใช้กันระหว่างอุปกรณ์ หรือจุดต่อจุด (Point-to-Point) ส่วน RS-422A นั้นเป็นมาตรฐานที่พัฒนามาจาก RS-232C ให้ได้ระยะทางไกลขึ้น และอัตราการสื่อสารมากขึ้น แต่ยังเป็นการสื่อสารข้อมูลระหว่างอุปกรณ์หนึ่งไปยังอุปกรณ์อื่น ๆ ได้อัตราสูงสุด 10 เท่าตัว ไม่สามารถส่งย้อนกลับจากอุปกรณ์ 10 ตัวได้ หรือกล่าวได้ว่าอุปกรณ์การสื่อสารข้อมูลตามมาตรฐาน RS-422A นั้นเป็นการสื่อสารข้อมูลแบบซิมเพล็ก (Simplex) คือทิศทางการสื่อสารข้อมูลเป็นทางเดียวตลอดเวลา ดังนั้น ถ้าต้องการออกแบบระบบให้เป็น โครงข่ายข้อมูลก็ไม่สามารถที่จะออกแบบได้ จึงได้มีการพัฒนามาตรฐานการสื่อสารข้อมูลแบบใหม่เพื่อรองรับความต้องการนี้ คือมาตรฐาน RS-485 ซึ่งเป็นมาตรฐานที่อาศัยหลักการของสัญญาณแบบดิฟเฟอร์เรนเชียล (Differential) เช่นเดียวกับมาตรฐาน RS-422A แต่การสื่อสารข้อมูลได้ทั้งสองทิศทางในสายสัญญาณเพียงคู่เดียว ซึ่งการสื่อสารข้อมูลแบบฮาฟดูเพล็ก (Half-Duplex) จากผลการทดลองการใช้สัญญาณในลักษณะดิฟเฟอเรนชียล (Differential) นี้จะทำให้ระยะทางและความเร็วในการสื่อสารข้อมูลเช่นเดียวกับมาตรฐาน RS-422A แต่มาตรฐาน RS-485 สามารถที่จะสื่อสารกันระหว่างอุปกรณ์ทั้งการรับและการส่งได้หลายจุดที่เรียกว่า มัลติพ้อยท์ คอมมิวนิเคชัน (Multipoint communication) โครงสร้างในการสื่อสารข้อมูลแบบ RS-485 ดังแสดงดังรูป



รูป 2.9 โครงสร้างการสื่อสารข้อมูลแบบอนุกรมตามมาตรฐาน RS-485

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ตารางที่2.4 ตารางเปรียบเทียบมาตรฐานการสื่อสารข้อมูลของ EIA

พารามิเตอร์ (Parameter)	RS-232C	RS-432-A	RS-422A	RS-485
โหมดการทำงาน	Single-ended	Single-ended	Differential	Differential
ความยาวของคู่สายและตัวส่งที่รับได้	1 ตัวส่ง 1 ตัวรับ	1 ตัวส่ง 10 ตัวรับ	1 ตัวส่ง 10 ตัวรับ	32 ตัวส่ง 32 ตัวรับ
ความยาวของคู่สายสัญญาณรับส่งข้อมูล	50 ฟุต	4000 ฟุต	4000 ฟุต	4000 ฟุต
อัตราการส่งข้อมูลสูงสุด (บิตต่อวินาที)	20 k	100 k	10 M	10 M
แรงดันไฟฟ้าโหมดร่วมสูงสุด	±2.5V	±6V	6V -2.5V	12V -7V
Driver load	±5V ต่ำสุด ±15V สูงสุด	±3.6V ต่ำสุด ±6.0V สูงสุด	±2V ต่ำสุด	±1.5V ต่ำสุด
Driver load Ω	3 k ถึง 7 k	450 ต่ำสุด	100 ต่ำสุด	60 ต่ำสุด
Drive slew rate	3 K/uS สูงสุด		NA	NA
กระแสลิมิตเมื่อกระแสถัดวงจร	500 mA ถัดวงจรกับ VCC หรือ GND	150 mA ถัดวงจรกับ GND	150 mA ถัดวงจรกับ GND	150 mA ถัดวงจรกับ GND 150 mA ถัดวงจรกับ 8V กับ 12V
ความต้านทานเอาท์พุท (output) ของตัวส่ง Ω	NA – power on 300 – power off	NA – power on 60K – power off	NA – power on 60K – power off	120 K power on, off
ความต้านทานอินพุท (input) ของตัวรับ Ω	3K ถึง 7K	4K	4K	12K
ความไวของตัวรับ	±3V	±200 mA	±200 mA	±200 mA

คุณสมบัติของการสื่อสารข้อมูลแบบ RS-485

คุณสมบัติของการสื่อสารข้อมูลแบบ RS-485ที่แตกต่างจาก RS-422A

คุณลักษณะเฉพาะของตัวส่ง RS-485

ตัวส่ง 1 ตัว สามารถที่จะขับโหลด (Load) ได้สูงสุด 32 ตัว โดยที่โหลด (Load) 1 ชุด ประกอบด้วยตัวส่ง 1 ตัวและตัวรับหนึ่งตัวและค่าของความต้านทานที่ตกคร่อมระหว่างคู่สายสัญญาณมีค่า 60Ω

- เอาท์พุท (Output) ของตัวส่งในสภาวะออฟ (Off) มีกระแสรั่วไหลไม่เกิน 100uA ในช่วงแรงดันไฟฟ้า โหมดร่วมระหว่าง –7 ถึง +12V
- เอาท์พุท (Output) ของตัวส่งให้แรงดันไฟฟ้าเอาท์พุท 1.5V ถึง 5V ในช่วงแรงดันไฟฟ้าร่วมระหว่าง –7 ถึง +12V

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

42663

- ตัวส่งมีวงจรป้องกันตัวเองที่ส่วนเอาท์พุท (Output) ในกรณีที่ตัวส่งหลาย ๆ ส่งข้อมูลออกมาพร้อม ๆ กัน

คุณลักษณะเฉพาะของตัวรับ RS-485

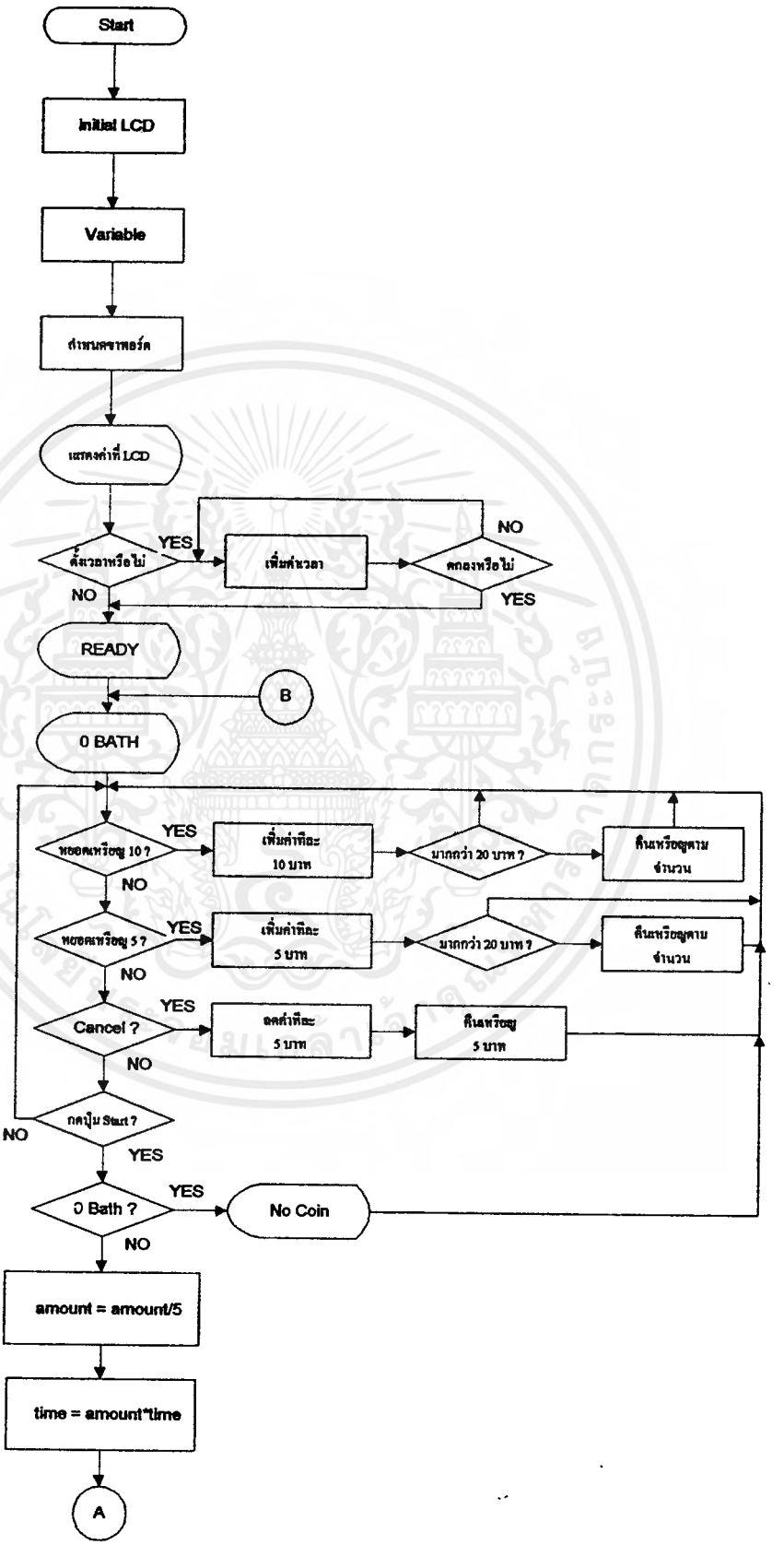
- ค่าความต้านทานอินพุท (Input) มีค่าสูง โดยมีค่าไม่น้อยกว่า 12 กิโลโอห์ม
- ตัวรับมีค่าแรงดันไฟฟ้าอินพุทโหมดร่วม ระหว่าง ค่า –7V ถึง +12V ตัวรับสามารถตอบสนองต่อสัญญาณที่แตกต่างกันจากสัญญาณที่โหมดร่วมได้ $\pm$200 mA (น้อยที่สุด)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 การทำงานของเครื่อง

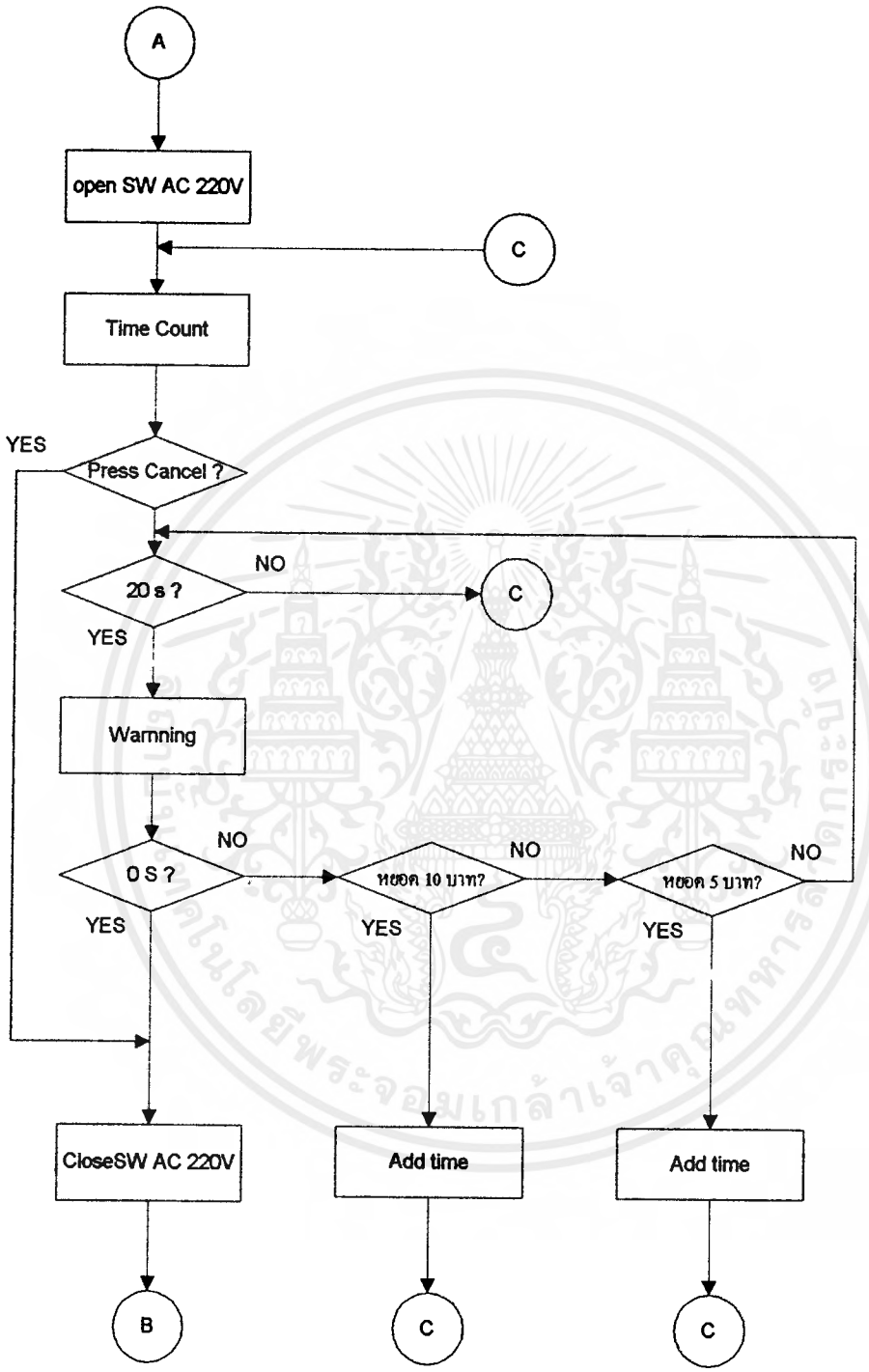
1. จอแสดงผลจะแสดงข้อความ WECOME TO COINS COUNTER และ SET TIME Y OR N
2. หน้าจอจะถามว่า จะ SET TIME หรือเปล่า ถ้าหากกด N เครื่องจะแสดงจำนวนเหรียญที่จะหยอดทันทีไม่มีการเซ็ตเวลา ถ้ากด Y จะต้องทำการตั้งเวลาต่อเหรียญว่าเราต้องการ 1 เหรียญต่อกี่นาทีแล้วแต่ผู้ใช้
3. ตั้งเวลาเสร็จแล้วก็ให้หยอดเหรียญตามความต้องการว่าจะเป็นเหรียญ 5 บาท หรือเหรียญ 10 บาท ในเครื่องนี้ได้เซ็ตไว้แค่หยอดได้ไม่เกิน 20 บาท ถ้าหยอดเกินเครื่องจะทำการทอนให้โดยจะทอนเฉพาะเหรียญ 5 บาท เท่านั้น
4. เมื่อหยอดเหรียญเสร็จก็ให้กด START ถ้ายังไม่ได้หยอดเหรียญ แล้วกดปุ่ม START จอแสดงผลจะแสดงข้อความ NO COINS จะไม่สามารถทำงานได้
5. เมื่อกด START แล้ว เครื่องจะทำการนับเวลาโดยจอแสดงผลจะแสดงข้อความ COUNT UP TO TIME
6. เมื่อเวลาหมดเครื่องจะทำการส่งสัญญาณเตือนว่าเวลาหมดแล้วและจะมีเสียงสัญญาณนับถอยหลังประมาณ 30 วินาที พร้อมจอแสดงผลจะแสดงข้อความ TO BE CONTINUE ถ้าไม่หยอดเหรียญเพิ่มก็จะจบการทำงาน แต่ถ้าหยอดเหรียญเพิ่มก็จะทำงานต่อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.10 ก Flowchart การทำงานของเครื่อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



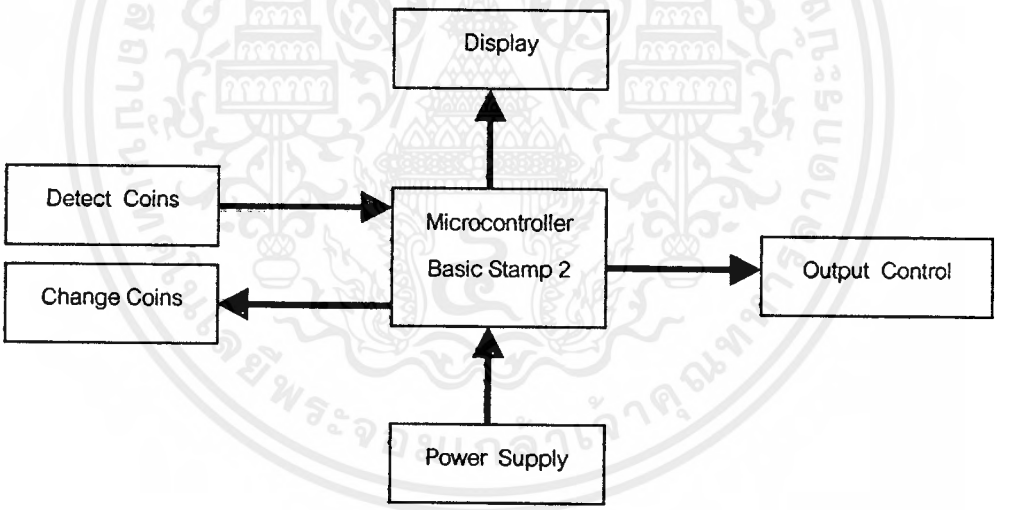
รูปที่ 2.10 ข Flowchart การทำงานของเครื่อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

การออกแบบ

โครงงานเครื่องตั้งเวลาหยอดเหรียญนี้ การออกแบบทั้งหมดเราเริ่มจากคิดหลักการทั้งหมดของเครื่องว่าน่าจะมีส่วนประกอบอะไรบ้าง ที่เราสามารถนำมาประยุกต์ใช้ หรือดัดแปลง และเราสามารถนำอุปกรณ์ และวงจรอะไรที่สามารถทำงานได้ตามที่เราต้องการ มาประกอบในเครื่องเราบ้าง โดยภาคไมโครคอนโทรลเลอร์ หรือ ภาคคอนโทรลเลอร์ เราจะนำมาควบคุมการทำงานทั้งหมดของเครื่องรวมทั้ง ควบคุมภาคกลไกด้วย และ ภาคกลไกก็จะทำหน้าที่ควบคุมเหรียญที่ผู้ใช้หยอดเข้ามาในเครื่อง โดยมีการทำงานต่าง ๆ เช่น คืนเหรียญ แล้วก็เก็บเหรียญเข้าที่เก็บเหรียญ โดยในตอนแรกเราได้ออกแบบมาเป็นบล็อกไดอะแกรม (Blockdiagram) คร่าว ๆ ดังนี้



รูปที่ 3.1 บล็อกไดอะแกรมรวม

จากบล็อกไดอะแกรมของเครื่องจะมีภาคต่าง ๆ ที่ทำงานดังต่อไปนี้

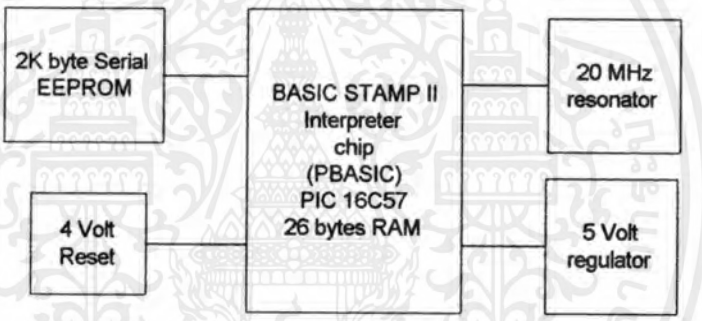
- ภาคไมโครคอนโทรเลอร์ (Control)
- ภาคตรวจนับจำนวนเหรียญ (Detect coins)
- ภาคควบคุมอุปกรณ์ระบบกลไก และภาคทอนเหรียญ (Change coins)
- ภาคจ่ายไฟ (Power supply)
- ภาคแสดงผล (Display)
- ภาคเอาท์พุท (Output)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

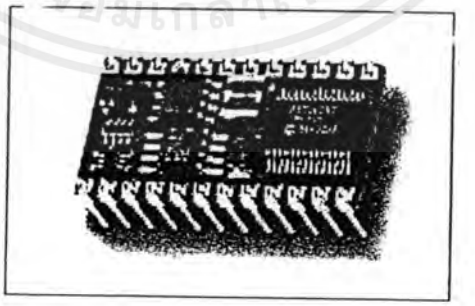
3.1 ภาคไมโครคอนโทรลเลอร์ (Control)

ซึ่งในภาคนี้การออกแบบก็จะเริ่มจาก การที่เราต้องเลือกว่าจะใช้ไมโครคอนโทรลเลอร์ตระกูลอะไรเบอร์อะไรมาเป็นตัวควบคุม หลักในภาคนี้ ซึ่งเราได้ใช้ไอซีไมโครคอนโทรลเลอร์ ในตระกูล BASIC STAMP รุ่น BS2-IC มีหน่วยความจำ 2 ชนิดอยู่บนตัวมัน ชนิดแรกคือ หน่วยความจำแรมขนาด 26 ไบต์ และหน่วยความจำอีอีพรอม ขนาด 2 กิโลไบต์ สามารถโปรแกรมใหม่ได้ 10 ล้านครั้ง และทำงานที่แรงดันไฟตรง 9-12V ใช้ความถี่ของสัญญาณนาฬิกา (clock) 20 MHz มีพอร์ตอินพุท เอาท์พุท 16 พอร์ต (P0-P15) และขาพอร์ตสำหรับสื่อสารข้อมูลอนุกรมอีก 2 ขา

ซึ่งในภาคนี้มีบล็อคไดอะแกรมดังนี้คือ



รูปที่ 3.2 บล็อกไดอะแกรมภาคไมโครคอนโทรลเลอร์



รูปที่ 3.3 ลักษณะภายนอกของ BS2-IC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากบล็อกไดอะแกรมดังแสดงในรูปที่ 3.2 จะเห็นว่าในภาคนี้จะประกอบไปด้วยไอซีหลายตัวซึ่งแต่ละตัวจะมีหน้าที่ต่าง ๆ ดังนี้คือ

3.1.1 PIC16C57 PBASIC 2 Interpreter Chip

เป็นไมโครคอนโทรลเลอร์เบอร์ PIC16C57 ซึ่งภายในบรรจุโปรแกรมแปลงภาษาเบสิก หรือ BASIC-2 อินเตอร์พรีตเตอร์เอาไว้ ไม่สามารถเข้ามาทำการแก้ไขได้ จะทำงานด้วยสัญญาณนาฬิกาความถี่ 20 MHz (20 MHz Resonator)

3.1.2 2048-byte Erasable Memory Chip (24LC16B)

เป็นไอซีหน่วยความจำอีอีพรอมอนุกรมเบอร์ 24LC16B มีความจุ 2 กิโลไบต์ (2KB) สามารถบรรจุคำสั่งของ PBASIC-2 ที่ใช้ในการรันโปรแกรมของ BS2 ได้ 500 คำสั่ง เก็บรักษาข้อมูลได้ยาวนาน คือเก็บได้นาน 10 ปี โปรแกรมใหม่ได้ 10 ล้านรอบ

3.1.3 5Volt Regulator (S-81350HG)

เป็นไอซีเรกูเลเตอร์ ทำหน้าที่ควบคุมให้เกิดแรงดันคงที่ +5V เพื่อจ่ายให้แก่วงจรทั้งหมดของ BS2

3.1.4 4Volt Reset (S-8054H)

เป็นไอซีวงจรรีเซตของ BS2 ทำหน้าที่สร้างสัญญาณรีเซตให้แก่ ไมโครคอนโทรลเลอร์ PIC16C57 โดยป้อนสัญญาณรีเซตเข้าที่ขา MCLR ของ PIC16C57 ในกรณีที่แรงดันไฟเลี้ยง +5V เกิดลดต่ำลงกว่า +4V ไม่ว่าจะด้วยสาเหตุของแบตเตอรี่ที่ใช้เป็นแหล่งจ่ายไฟมีระดับแรงดันลดลง หรือวงจรแหล่งจ่ายไฟภายนอกของ BS2 เกิดลัดวงจรทำให้แรงดันต่ำลง IC3 ก็จะทำงานสร้างสัญญาณรีเซตให้แก่ไมโครคอนโทรลเลอร์ PIC16C57 เพื่อควบคุมให้อยู่ในสภาวะรีเซตจนกว่าแรงดันไฟเลี้ยงของ BS2 จะกลับมาอยู่ในสภาวะปกติ

3.2 ภาคตรวจนับจำนวนเหรียญ (Detect coins)

ในภาคนี้จะทำการตรวจนับจำนวนเหรียญที่ผู้ใช้ทำการหยอดเข้ามา โดยจะใช้หยอดเข้ามาโดยใช้ไมโครสวิตซ์ (Micro Switch) เป็นตัวตรวจสอบ ซึ่งจะทำการนับจำนวนเหรียญที่เข้ามาแล้วแสดงผลออกมาทาง LED ตัวเลข 7 ส่วน

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 ภาคควบคุมอุปกรณ์ระบบกลไกและภาคทอนเหรียญ (Change Coins)

ในภาคนี้จะทำหน้าที่ไปควบคุมระบบกลไกของเครื่องตั้งเวลาหยอดเหรียญนี้โดยเราจะใช้โซลีนอยด์ (Solinoid) ซึ่งการทำงานของตัวโซลีนอยด์จะทำงาน โดยเมื่อมีการจ่ายไฟให้กับตัวโซลีนอยด์ ตัวโซลีนอยด์จะทำการดึงตัวมันเอง ซึ่งเราใช้หลักการนี้มาใช้ทำตัวคัดเหรียญทิ้ง เมื่อเราจ่ายไฟให้กับโซลีนอยด์ ตัวโซลีนอยด์จะมีการดึงตัวเองเกิดขึ้น ทำให้เราสามารถคัดเหรียญไปเก็บไว้หรือให้ทอนเหรียญได้

3.4 ภาคจ่ายไฟ (Power Supply)

ภาคนี้จะทำการจ่ายไฟให้กับวงจรทุก ๆ ภาคที่มีอยู่ในเครื่องนี้ ซึ่งในที่นี้เราใช้ไฟดีซี (DC) ขนาด 12 โวลท์ จ่ายให้กับภาคไมโครคอนโทรลเลอร์ และขนาด 5 โวลท์ ซึ่งเราจะจ่ายให้กับวงจรที่เหลือ เราใช้วงจรเร็กกูเลทในการสร้างแรงดันกระแสตรงขนาด 5 โวลท์ จากแหล่งจ่ายไฟฟ้ากระแสสลับที่เข้ามาขนาด 220 โวลท์ จากแหล่งจ่ายไฟฟ้ากระแสสลับทั่วไปตามบ้านเรา

3.5 ภาคแสดงผล (Display)

ในการแสดงผลของเครื่องตั้งเวลาหยอดเหรียญนี้เราจะใช้หน้าจอแสดงผลแบบ LCD ซึ่งจะใช้แสดงข้อความและใช้ LED ตัวเลข 7 ส่วน แบบ 4 หลัก เป็นตัวแสดงเวลาและใช้ LED ตัวเลข 7 ส่วน แบบ 2 หลัก 3 ชุด ซึ่งชุดแรกจะแสดงจำนวนเวลาต่อ 1 เหรียญ ชุดที่สองจะแสดงจำนวนของเหรียญ 10 และชุดที่สามจะแสดงจำนวนของเหรียญ 5

3.6 ภาคเอาท์พุท (Output)

เป็นภาคที่ไว้สำหรับจ่ายไฟ 220 V AC 50 HZ ให้กับอุปกรณ์ภายนอก ซึ่งในการควบคุมเวลาในการจ่ายไฟจะขึ้นอยู่กับภาคคอนโทรล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
การพัฒนาการเขียนโปรแกรม

ในการพัฒนาการเขียนโปรแกรมเพื่อควบคุมการทำงานของเบสิกแสตมป์ 2 นั้นจะใช้ภาษาเบสิกที่เรียกว่า พีเบสิก (PBASIC) โดยผ่านโปรแกรมที่ชื่อว่า เบสิกแสตมป์ เอดิเตอร์ (BASIC Stamp Editor) โปรแกรมนี้จะถูกใช้ในการติดต่อคอมพิวเตอร์กับเบสิกแสตมป์ 2 ผ่านทางพอร์ตอนุกรม ใช้ในการเขียนและแก้ไขโปรแกรมภาษาเบสิกหรือทำหน้าที่เป็นเอดิเตอร์นั่นเอง โดยโปรแกรมจะทำการตรวจสอบการเขียนโปรแกรมลงบนตัวเบสิกแสตมป์เพื่อทำการรัน

โดยเราจะทำการเขียนโปรแกรมและทดสอบกับอุปกรณ์ต่าง ๆ ที่จะนำมาใช้ในเครื่องต้นแบบโดยเราจะทำการต่อวงจรต่าง ๆ ลงบนบอร์ดทดลอง NX-BS2 ซึ่งบอร์ดทดลองนี้ได้จัดเตรียมอุปกรณ์ต่าง ๆ ตลอดจนการติดต่อกับคอมพิวเตอร์เอาไว้อย่างครบถ้วนสมบูรณ์ อาทิเช่น การทดลองขับโมดูล LCD การติดต่อระหว่างขาพอร์ตของ BS2 กับสวิตซ์ เป็นต้น

ไอซี BS2 นี้เราสามารถทำการโปรแกรมลงไปในตัวไอซีได้ถึง 10 ล้านรอบ ดังนั้นเราสามารถเขียนโปรแกรมแล้วนำมาทดสอบการทำงานเป็นภาค ๆ แล้วทำการทดลองทีละวงจรทีละภาคได้แล้วจึงทำการทดสอบทุกวงจรโดยรวมอีกครั้งหนึ่ง เพื่อทดสอบการทำงานร่วมกันทั้งหมดว่าสามารถทำงานร่วมกันได้ ในการเขียนโปรแกรมเราทำการเขียนโดยใช้คอมพิวเตอร์แล้วเมื่อได้โปรแกรมที่ทำการทดสอบโดยการทดสอบในโปรแกรมของเครื่องคอมพิวเตอร์แล้วก็ทำการโปรแกรมที่ทำงานแล้วลงไปในตัวไอซีแล้วจึงทำการนำไปทดลองกับตัววงจรใช้งานจริงอีกที

4.1 โปรแกรมการใช้งาน

ในตัวโปรแกรมการใช้งาน จะประกอบไปด้วยการทำงานดังนี้ โดยเริ่มแรก หน่วยประมวลผลจะทำการเซตค่าเริ่มต้นต่าง ๆ และทำการกำหนดค่าตัวแปรให้กับระบบ และทำการกำหนดขาพอร์ต จากนั้นจะทำการแสดงข้อมูลต่าง ๆ ออกมาทางหน้าจอแสดงผลคือ จอ LCD โดยจะแสดงข้อความต่าง ๆ ออกมา และค่าของจำนวนเหรียญ ซึ่งได้กำหนดไว้หยอดเหรียญได้ไม่เกิน 20 บาท ถ้าเกินจะทำการทอนเป็นเหรียญ 5 บาท โดยโปรแกรมจะสั่งงานให้ไปควบคุมวงจร ที่ควบคุมโซลีนอยด์วาล์ว ให้ทำงานเก็บเหรียญ หรือ ทำการทอนเหรียญ

ผู้ใช้ยังสามารถที่จะโปรแกรมตั้งค่าเวลาต่อจำนวนเหรียญได้ว่าผู้ใช้ต้องการ กี่เหรียญหรือกี่บาทต่อกี่นาที และโปรแกรมจะทำการคำนวณออกมาเป็นค่าเวลารวมแล้วเริ่มนับ เวลารวมแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผู้ใช้ยังสามารถที่จะโปรแกรมตั้งค่าเวลาต่อจำนวนเหรียญได้ว่าผู้ใช้ต้องการ กี่เหรียญหรือกี่บาทต่อกี่นาที และโปรแกรมจะทำการคำนวณออกมาเป็นค่าเวลารวมแล้วเริ่มนับ เวลารวม แล้วเริ่มนับถอยหลังเมื่อมีการรับคำสั่งเข้ามาว่าผู้ใช้ตอบตกลงการทำงานของเครื่อง เมื่อหมดเวลาก็จะมีการเตือนให้หยอดเหรียญเพิ่ม ถ้าไม่หยอดเพิ่มก็จะจบการทำงาน ถ้าหยอดเพิ่มโปรแกรมจะทำการนับเวลาถอยหลังต่อไป

4.2 ชุดโปรแกรมควบคุมการทำงานของไมโครคอนโทรลเลอร์

ในโปรแกรมชุดนี้จะมีโปรแกรมที่เราสามารถตั้งค่าเวลาต่อจำนวนเหรียญได้ คือ ในจำนวนเหรียญที่หยอดเข้ามา 1 เหรียญ เราสามารถโปรแกรมตั้งค่าเวลาต่อเหรียญได้ และจะมีการคำนวณออกมาเป็นค่าเวลารวม แล้วทำการเริ่มนับเวลาถอยหลังเมื่อมีการรับคำสั่ง เข้ามาว่าผู้ใช้ตอบตกลงการทำงานของเครื่อง

4.3 ชุดโปรแกรมควบคุมการตรวจจับเหรียญ

ในโปรแกรมการทำงานชุดนี้จะมีการนับจำนวนเหรียญที่เข้ามาโดยผ่านจากไมโครสวิตซ์แล้วจะมีการนับจำนวนเหรียญให้ได้ 20 บาท เมื่อมีการนับครบ 20 บาทแล้ว ถ้ามีการหยอดเหรียญเพิ่มจะทำการทอนเหรียญ โดยถ้าเป็นเหรียญ 5 บาท โปรแกรมจะสั่งงานให้ไปควบคุมวงจรที่ควบคุมโซลีนอยด์ให้ทำการทอนเหรียญ 5 บาท จำนวน 1 เหรียญ แต่ถ้าเป็น เหรียญ 10 บาท โปรแกรมจะสั่งงานให้ไปควบคุมวงจรที่ควบคุมโซลีนอยด์ให้ทำการทอนเหรียญ 5 บาท จำนวน 2 เหรียญ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
การทดลองและวิธีการใช้งาน

การทำงานของเครื่อง

ในการทำงานของเครื่องหยอดเหรียญ จะสามารถรับเหรียญได้ 2 ขนาด คือ เหรียญ 5 บาท และเหรียญ 10 บาท โดยเครื่องหยอดเหรียญนี้จะมี SWITCH ในการควบคุมทั้งหมดอยู่ 7 ปุ่มด้วยกันคือ

1. **START** จะเป็นปุ่มที่ใช้เริ่มต้นการทำงาน
2. **CANCEL** เป็นปุ่มสำหรับการคืนเหรียญและหยุดการทำงานของเครื่อง
3. **RESET** เป็นปุ่มเริ่มต้นการทำงานใหม่ทั้งหมดของเครื่อง
4. **YES** เป็นปุ่มเพื่อเข้าสู่โหมดการตั้งเวลา
5. **SET** เป็นปุ่มสำหรับการตั้งเวลา
6. **OK** เป็นปุ่มเพื่อการกำหนดเวลาและเข้าสู่โหมดปรกติ
7. **NO** เป็นปุ่มเพื่อเลือกเข้าสู่โหมดปรกติ

โดยการทำงานของเครื่องตั้งเวลาจะแบ่งออกเป็นการทำงานอยู่ 2 โหมดคือ

- การทำงานในโหมดปรกติ
- การทำงานในโหมดการตั้งเวลา

5.1 การทำงานในโหมดปรกติ

การทำงานในโหมดปรกตินี้ จะเป็นการทำงาน โดยจะทำงานได้ก็ต่อเมื่อมีการหยอดเหรียญให้กับเครื่องตั้งเวลาหยอดเหรียญ และมีการกดปุ่ม Start เครื่องก็จะทำงานตามปรกติ

5.1.1 วิธีการเข้าสู่โหมดปรกติ

วิธีการเข้าสู่โหมดปรกติ จะมีอยู่ 2 วิธีด้วยกันคือ

5.1.1.1 เมื่อเปิดเครื่อง ที่ LCD จะปรากฏข้อความว่า " SET TIME Y OR N " ให้ทำการกดปุ่ม NO เครื่องก็จะเข้าสู่โหมดการทำงานปรกติ

5.1.1.2 เมื่อเปิดเครื่อง เครื่องจะเข้าสู่โหมดการทำงานปรกติเองโดยอัตโนมัติ โดยนับหลังจากการเปิดเครื่อง 10 วินาที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รายละเอียดในการทำงานในโหมดปรกติ

ในการทำงานโหมดปรกติมีรายละเอียดดังนี้

1. เมื่อมีการเข้าสู่การทำงานในโหมดปรกติแล้ว จะมีข้อความปรากฏที่ LCD ว่า "INSERT COINS" และเครื่องจะรอรับเหรียญ เมื่อมีการหยอด โดยการหยอดเหรียญนั้น สามารถหยอดได้ทั้งเหรียญ 5 และเหรียญ 10 บาทได้

2. เมื่อมีการหยอดเหรียญแล้วเครื่องจะทำการแสดงจำนวนเงินที่หยอดลงไปในเครื่องหยอดเหรียญที่ LCD เช่นหยอดเหรียญ 10 บาท ก็จะปรากฏข้อความที่ LCD ว่า "10 BATH"

3. การทำงานของเครื่องจะรับจำนวนเงินสูงสุดได้ 20 บาท และหากมีการหยอดเหรียญเกิน 20 บาทเครื่องจะทำการทอนเงิน (คืนเหรียญ) เป็นเหรียญ 5 บาท เท่ากับค่าเงินที่หยอดเกินไปนั้น เช่น หากในเครื่องมีจำนวนเงินอยู่ 20 บาทแล้ว และมีการหยอดเพิ่มอีก 5 บาท เครื่องจะทำการคืน คือ เหรียญ 5บาท จำนวน 1 เหรียญ แต่หากหยอดเพิ่มอีกเป็น 10 บาท เครื่องจะทำการคืน คือ เหรียญ 5 บาท จำนวน 2 เหรียญ

4. หากมีการกดปุ่ม CANCEL นั้น เครื่องจะทำการคืนเหรียญ 5 บาท จำนวน 1 เหรียญ และจะสามารถคืนได้ เท่ากับจำนวนเงินที่ค้างอยู่ในเครื่อง

5. ในการเริ่มการทำงานนั้น จะต้องทำการกดปุ่ม START ซึ่งหากว่า ไม่มีการหยอดเหรียญก่อนหน้านี้ หรือ มีจำนวนเงินเป็น 0 บาท และทำการกดปุ่ม START แล้ว เครื่องจะแจ้งว่า ไม่มีจำนวนเงินอยู่ในเครื่อง คือ จะปรากฏข้อความที่ LCD ว่า " NO COINS" และจะส่งเสียงเตือนด้วย เมื่อปรากฏข้อความดังกล่าวแล้ว สักพัก ก็จะปรากฏ ข้อความว่า " INSERT COINS " เพื่อรอรับเหรียญต่อไป

6. หากมีการกดปุ่ม START โดยที่มีจำนวนเงินอยู่ในเครื่องแล้ว เครื่องก็จะสามารถทำงานได้ และส ามารถจ่ายแรงดันไฟฟ้าที่ภาคเอาท์พุทได้นานตามระยะเวลาที่ได้ตั้งเวลาและจำนวนเงินที่หยอดลงไป และจะแสดงข้อความที่ LCD ว่า " COUNT UP TO TIME "

7. ในระหว่างการทำงานนั้นสามารถที่จะกดปุ่ม CANCEL ได้ หากมีการกดปุ่ม CANCEL แล้วเครื่องจะหยุดจ่ายแรงดันที่ภาคเอาท์พุทและจะกลับไปสู่หน้าจอเดิมคือ " INSERT COINS" เพื่อรอรับเหรียญและทำงานครั้งต่อไป

8. เมื่อเครื่องทำงานและหมดเวลาตามจำนวนเวลาที่ได้คำนวณไว้แล้วนั้น เครื่องจะส่งเสียงเตือน และจะแสดงข้อความที่ LCD คือ "TO BE CONTINUED" เพื่อรอรับเหรียญ 5 บาท หรือ 10 บาท เพื่อที่จะสามารถเพิ่มเวลาต่อได้ตามจำนวนเหรียญที่หยอดลงไปนั้น โดยมีเวลาในการที่จะสามารถหยอดเหรียญเพิ่มนี้คือ 10 วินาที นับจากเริ่มมีเสียงเตือน และหากไม่มีการหยอดเหรียญเพิ่ม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เครื่องจะทำการตัดการจ่ายไฟที่ภาคเอาท์พุท และกลับเข้าสู่หน้าจอหลักคือ " INSERT COINS" เพื่อรอรับเหรียญ และการทำงานครั้งต่อไป

5.2 การทำงานในโหมดตั้งเวลา

การทำงานในโหมดการตั้งเวลานั้น จะเป็นการตั้งเวลาเป็น นาที / 5บาท โดยการตั้งเวลานั้นสามารถ ตั้งเวลาได้ตั้งแต่ 1 นาทีถึง 90 นาที มีขั้นตอนการตั้งเวลาดังนี้

1. เมื่อเปิดเครื่อง ที่ LCD จะปรากฏข้อความว่า " SET TIME Y OR N " ให้ทำการกดปุ่ม YES เครื่องก็จะเข้าสู่โหมดการตั้งเวลา
2. เมื่อเข้าสู่โหมดการตั้งเวลาแล้ว LCD จะปรากฏข้อความ " SETTING" และจะสามารถตั้งเวลาได้โดยการ กดปุ่ม SETTIME โดยจะแสดงผลของเวลาที่ตั้งโดย 7- SEGMENT และจะเพิ่มค่าขึ้นทีละ 1 นาที เมื่อมีการกดปุ่ม SET
3. เมื่อได้เวลาที่ต้องการแล้ว ให้กดปุ่ม OK เครื่องจะรับทราบถึงการตั้งเวลา ต่อ เหรียญ และเข้าสู่โหมดการทำงานปรกติต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6
บทสรุปและแนวทางการพัฒนา

จากการทำงานของเครื่องตั้งเวลาหยอดเหรียญนี้มีอุปสรรคในการทำงานอยู่บ้างพอสมควร ซึ่งก็สามารถแก้ไขไปได้ด้วยดี ซึ่งปัญหาที่สามารถกล่าวถึงได้มีดังต่อไปนี้

1. ปัญหาอันเนื่องมาจากการเขียนโปรแกรมเพื่อใช้ในการตรวจสอบเวลาเพื่อให้ได้ตรงตามที่กำหนด เนื่องจากการทำงานของไมโครคอนโทรลเลอร์จะต้องทำงานเพื่อตรวจสอบเวลาและในขณะเดียวกันก็ต้องตรวจสอบอินพุทที่จะเข้ามาในขณะทำการตรวจสอบเวลาด้วยนั้น เป็นการยากที่จะสามารถกำหนดเวลาให้ตรงตามที่ออกแบบได้ ดังนั้นการแก้ปัญหาจะใช้วิธีการประมาณให้ได้ค่าที่ใกล้เคียงที่สุดที่พอรับได้
2. ปัญหาอันเนื่องมาจากการกระชากกระแส ของ โซลินอยด์ เนื่องจากโซลินอยด์ที่ใช้เป็นโซลินอยด์ที่ใช้ไฟ AC ในการควบคุมจึงมีปัญหาเรื่องการตัดต่อ โซลินอยด์เพื่อการคืนเหรียญจะเกิดการกระชากกระแสขึ้น โดยการแก้ปัญหานั้นจะต้องใช้ วาริสเตอร์ ค่าหนึ่งที่เหมาะสมคร่อมหน้าสัมผัสของรีเลย์ และ คาปาซิสเตอร์ ค่าหนึ่งที่เหมาะสมมาต่อคร่อมกับไมโครคอนโทรลเลอร์

แนวทางการพัฒนา

ในการพัฒนาเครื่องตั้งเวลาหยอดเหรียญนี้ หากสามารถพัฒนาในเรื่องของส่วนการคืนเหรียญให้มีประสิทธิภาพมากขึ้นให้สามารถใส่เหรียญเพื่อที่จะใช้ทอนได้มากขึ้นก็จะเป็นเรื่องที่ดีมาก เพราะเครื่องตั้งเวลาหยอดเหรียญนี้สามารถใส่เหรียญที่เอาไว้คืนได้สูงสุดเพียง 20 เหรียญเท่านั้น และในการพัฒนาในเรื่องนี้ย่อมหมายถึง การเปลี่ยนแปลงทางแมคคานิคด้วย ซึ่งในการปรับปรุงแก้ไขนั้นจะต้องอาศัยความรู้และประสบการณ์ด้วยจึงจะเป็นผลดีที่สุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

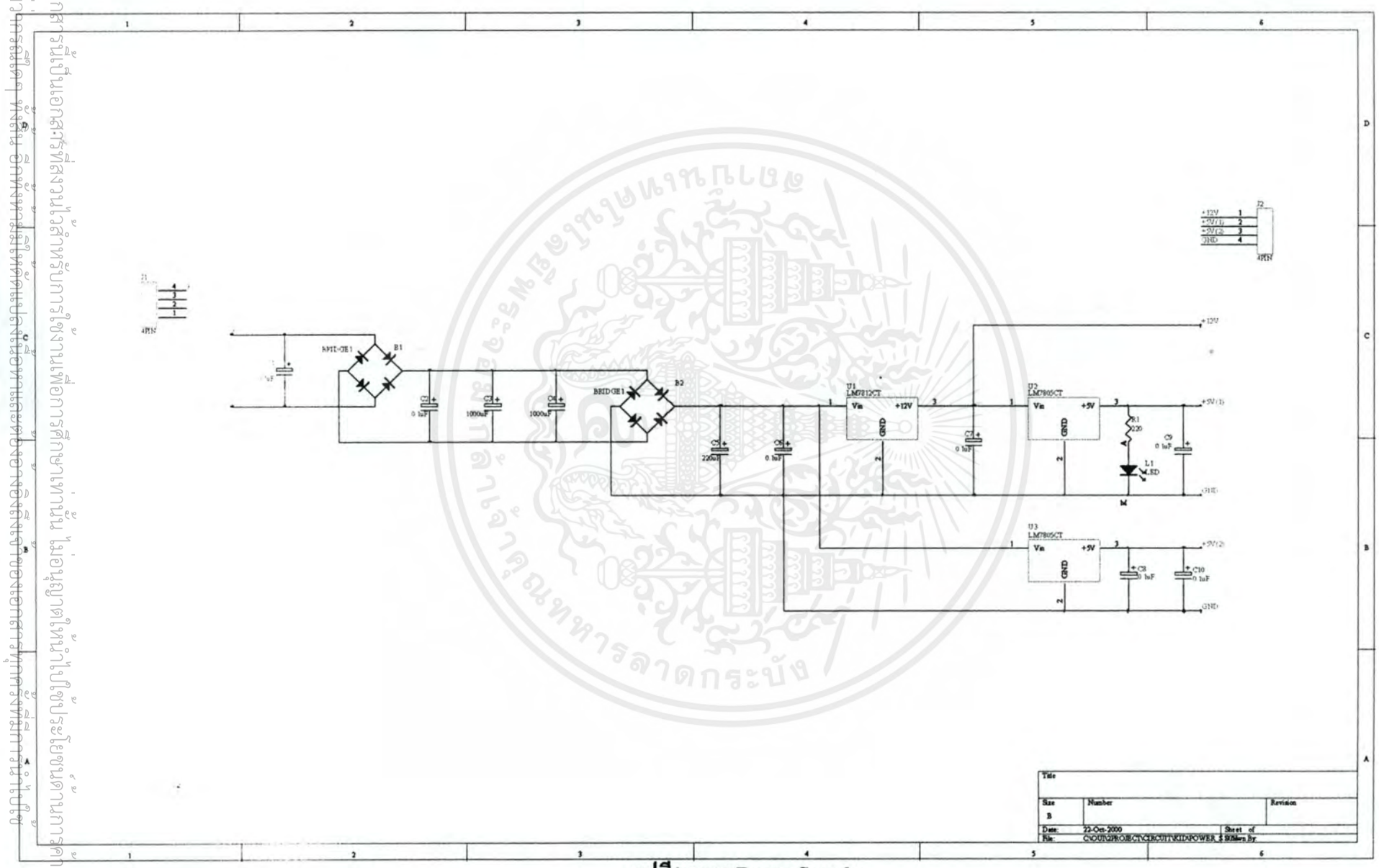
1. กฤษดา ไชเย็น. และคณะ วรพจน์ กรแก้ววัฒนกุล. ชัยวัฒน์ ลิ้มพรจิตรวิไล. **เรียนรู้ไมโครคอนโทรลเลอร์อย่างง่ายกับเบสิกแสตมป์ 2** : บริษัท อินโนเวตีฟ เอ็กเพอริเมนต์ จำกัด
2. http://www.allegromicro.com
3. http://www.parallaxinc.com
4. http://www.stampinclass.com

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

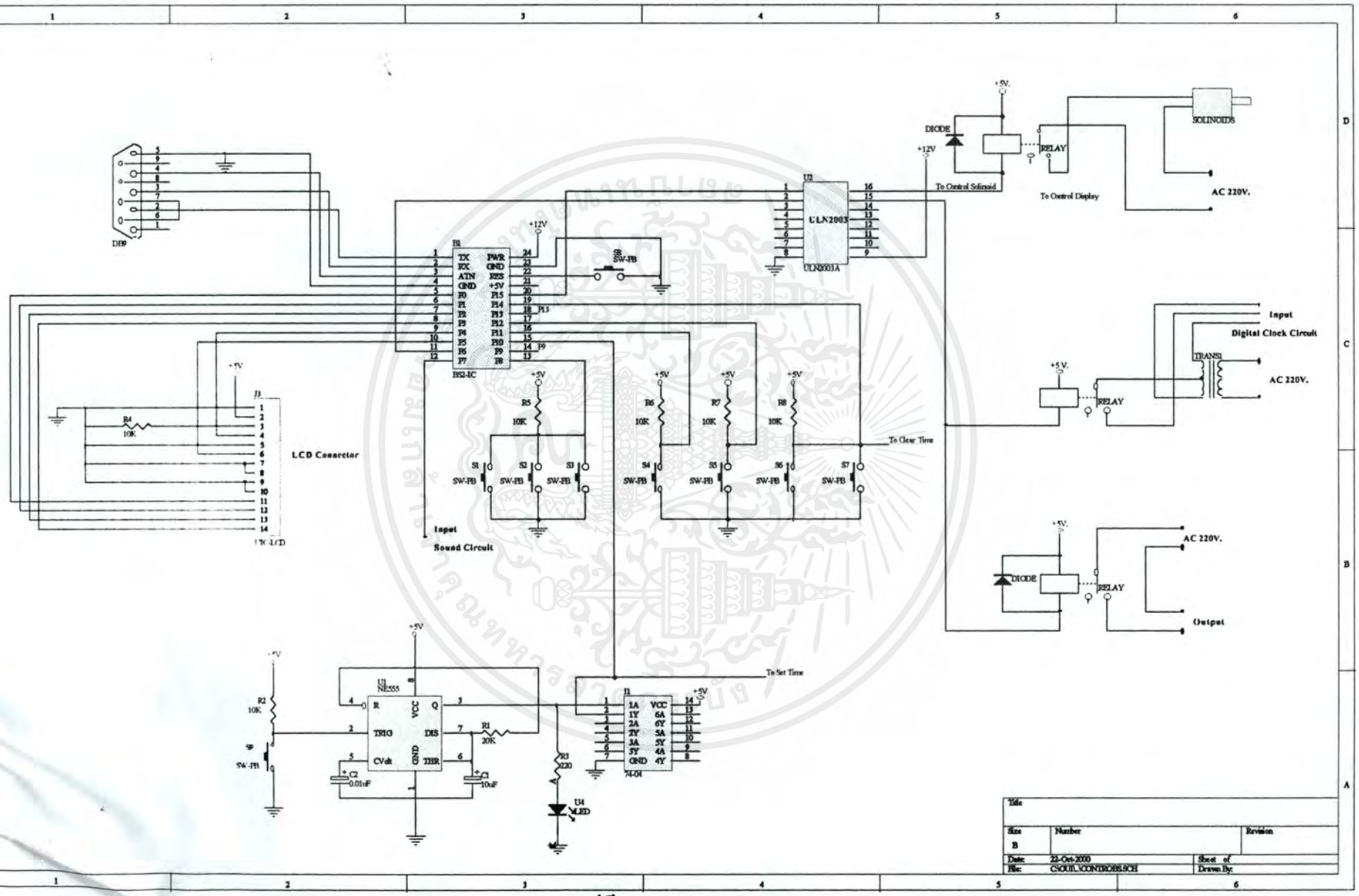
ภาคผนวก ก.

รายละเอียดทางฮาร์ดแวร์

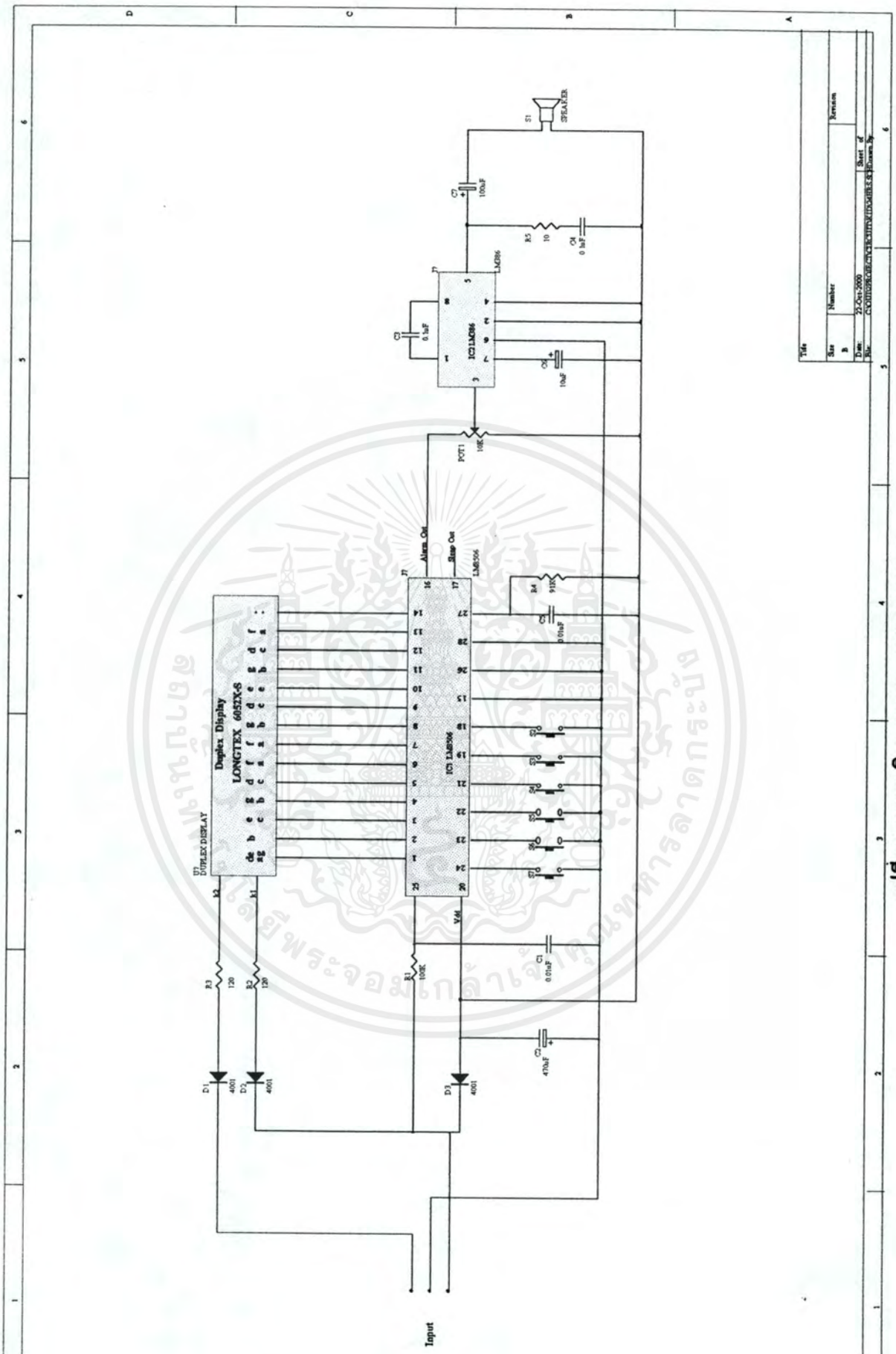
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่1 วงจร Power Supply

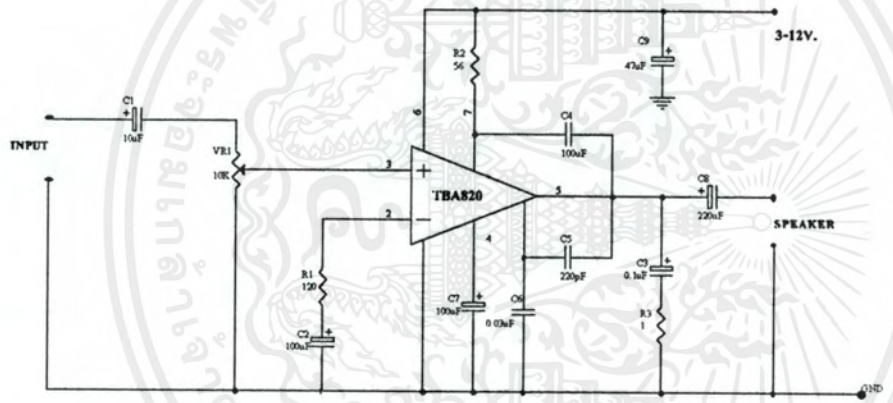


รูปที่2 วงจรภาค Control



รูปที่3 วงจรนาฬิกา 24 Hrs. Digital Clock

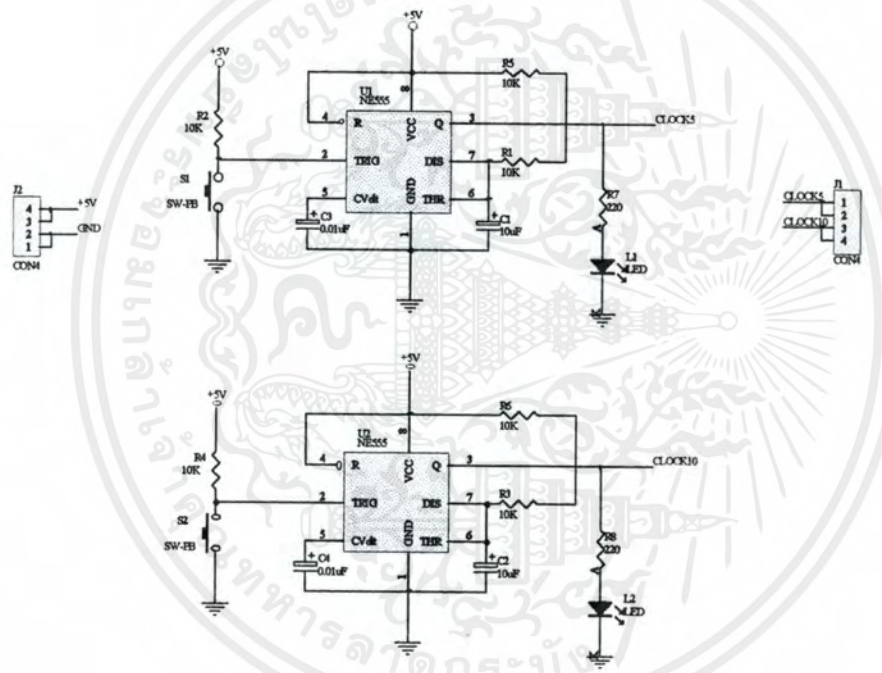
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Title		
Size: B	Number	Revision
Date: 22-Oct-2000		Sheet of
File: C:\OUTZPROJECT\CIRCUIT\VR1D\SPEAKER		Drawn By:

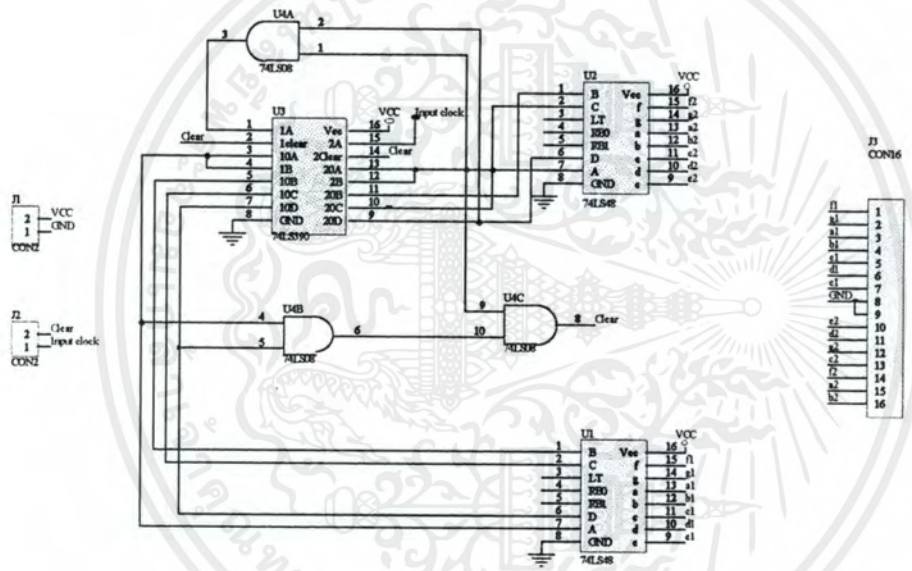
รูปที่4 วงจร ขยายเสียง

ลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีราชมงคลตามพระราชบัญญัติลิขสิทธิ์ พ.ศ. ๒๕๓๗ ห้ามมิให้ทำซ้ำ ดัดแปลง หรือนำไปใช้ในการค้า ไม่ว่ากรณีใดๆ เว้นแต่ได้รับอนุญาตเป็นลายลักษณ์อักษรจากมหาวิทยาลัยเทคโนโลยีราชมงคลเท่านั้น

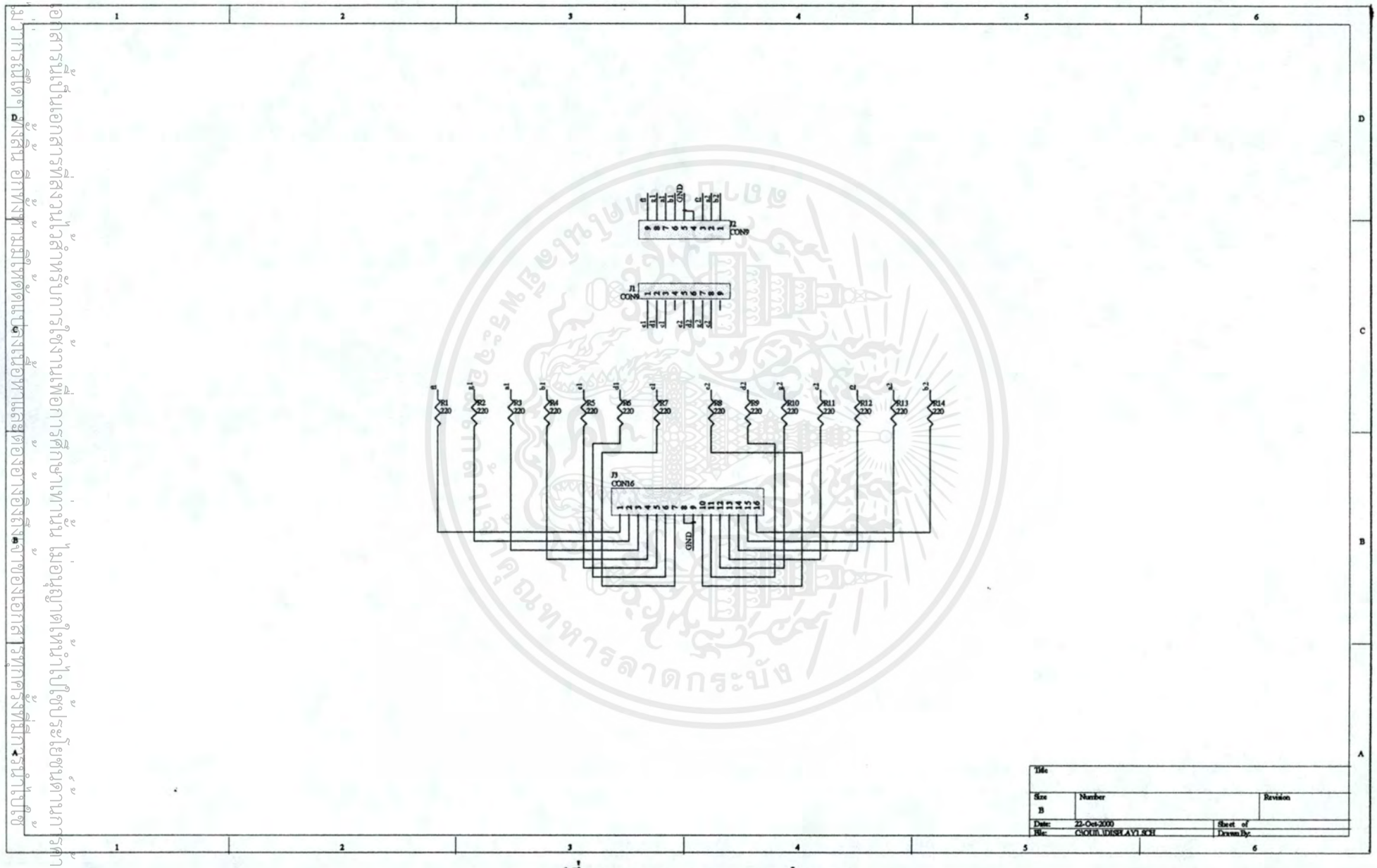


Title			
Size: B	Number		Revision
Date:	23-Oct-2000	Sheet of	
File:	C:\OUR_VCLOCK.SCH	Drawn By:	

รูปที่5 วงจร Detect Coins



รูปที่6 วงจรภาค Control ส่วนของ 7-Segment



รูปที่7 วงจรภาค Output ของส่วน 7-Segment

ภาคผนวก ข.
โปรแกรมใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
OUTPUT  15
RS CON 4
E   CON 5
CLRLCD  CON $01
DISPLF   CON $18
DISPRT   CON $14
DEL       VAR    WORD
WIT       VAR    WORD
ADD       VAR    WORD
TIH       VAR    BYTE
GOI       VAR    BYTE
REW       VAR    BYTE
AUY       VAR    BYTE
CHAR      VAR    BYTE
CHAR2 VAR        BYTE
WW        VAR    BYTE
OLD       VAR    NIB
REC       VAR    NIB
COUN      VAR    NIB
PBOT      VAR    NIB
PBOT2     VAR    NIB
PBOT3     VAR    NIB
PBOT4     VAR    NIB
PBOT5     VAR    NIB
CHK       VAR    BIT
GIB       VAR    BIT
INCHA     VAR    BIT
AMOUNT VAR BYTE
PICK      VAR    NIB
INDEX1 VAR       BYTE
INDEX2 VAR BYTE
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
REW = 1
GOI = 0
MSG0 DATA "  INSERT COINS  "
MSG1 DATA "   WELCOME TO   "
MSG2 DATA " COINS COUNTER  "
MSG3 DATA "    READY     "
MSG4 DATA "    5 BATH    "
MSG5 DATA "    10 BATH    "
MSG6 DATA "    15 BATH    "
MSG7 DATA "    20 BATH    "
MSG8 DATA "COUNTUP TO TIME "
MSG9 DATA " TO BE CONTINUE "
MSG10 DATA "   NO COINS    "
MSG11 DATA " SET TIME Y OR N"
MSG12 DATA "   SETTING     "

INIT:
   DIRL = %01111111
        OUTS = $0000
LCDINI:
        PAUSE 500
        OUTA = %0011
        PULSOUT E,1
        PAUSE 5
        PULSOUT E,1
        PULSOUT E,1
        OUTA = %0010
        PULSOUT E,1
        CHAR = %00101000
        GOSUB LCDCMD
        CHAR = %00001100
        GOSUB LCDCMD
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        CHAR = %00000110
        GOSUB LCDCMD
        CHAR = CLRLCD
        GOSUB LCDCMD
'BEGIN
        OUTPUT 13
        FREQOUT 7,350,4000
        CHAR = CLRLCD
        GOSUB LCDCMD
        FOR INDEX1 = 0 TO 7
        READ MSG1 + INDEX1,CHAR
        GOSUB LCDWR
        NEXT
        GOSUB LCDWR2
        FOR INDEX1 = 8 TO 15
        READ MSG1 + INDEX1,CHAR
        GOSUB LCDWR
        NEXT
        PAUSE 1000
        OUT13 = 1
        PAUSE 500
        OUT13 = 0
        FREQOUT 7,350,5000
        CHAR = CLRLCD
        GOSUB LCDCMD
        FOR INDEX1 = 0 TO 7
        READ MSG2 + INDEX1,CHAR
        GOSUB LCDWR
        NEXT
        GOSUB LCDWR2
        FOR INDEX1 = 8 TO 15
        READ MSG2 + INDEX1,CHAR
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                GOSUB LCDWR
                NEXT
                PAUSE 1000
                GOTO START2
READY:
CHAR = CLRLCD
                GOSUB LCDCMD
                FOR INDEX1 = 0 TO 7
                READ MSG3 + INDEX1,CHAR
                GOSUB LCDWR
                NEXT
                GOSUB LCDWR2
                FOR INDEX1 = 8 TO 15
                READ MSG3 + INDEX1,CHAR
                GOSUB LCDWR
                NEXT
                PAUSE 1000
                RETURN
'0
C0B:            CHAR = CLRLCD
                GOSUB LCDCMD
                FOR INDEX1 = 0 TO 7
                READ MSG0 + INDEX1,CHAR
                GOSUB LCDWR
                NEXT
                GOSUB LCDWR2
                FOR INDEX1 = 8 TO 15
                READ MSG0 + INDEX1,CHAR
                GOSUB LCDWR
                NEXT
                IF GIB = 0 THEN START
                GOTO START1
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
'1
C5B:    CHAR = CLRLCD
        GOSUB LCDCMD
        FOR INDEX1 = 0 TO 7
        READ MSG4 + INDEX1,CHAR
        GOSUB LCDWR
        NEXT
        GOSUB LCDWR2
        FOR INDEX1 = 8 TO 15
        READ MSG4 + INDEX1,CHAR
        GOSUB LCDWR
        NEXT
        GOTO START1
'2
C10B:   CHAR = CLRLCD
        GOSUB LCDCMD
        FOR INDEX1 = 0 TO 7
        READ MSG5 + INDEX1,CHAR
        GOSUB LCDWR
        NEXT
        GOSUB LCDWR2
        FOR INDEX1 = 8 TO 15
        READ MSG5 + INDEX1,CHAR
        GOSUB LCDWR
        NEXT
        GOTO START1
'3
C15B:   CHAR = CLRLCD
        GOSUB LCDCMD
        FOR INDEX1 = 0 TO 7
        READ MSG6 + INDEX1,CHAR
        GOSUB LCDWR
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                NEXT
                GOSUB LCDWR2
                FOR INDEX1 = 8 TO 15
                READ MSG6 + INDEX1,CHAR
                GOSUB LCDWR
                NEXT
                GOTO START1
'4
C20B:           CHAR = CLRLCD
                GOSUB LCDCMD
                FOR INDEX1 = 0 TO 7
                READ MSG7 + INDEX1,CHAR
                GOSUB LCDWR
                NEXT
                GOSUB LCDWR2
                FOR INDEX1 = 8 TO 15
                READ MSG7 + INDEX1,CHAR
                GOSUB LCDWR
                NEXT
                GOTO START1

CTT:
CHAR = CLRLCD
                GOSUB LCDCMD
                FOR INDEX1 = 0 TO 7
                READ MSG8 + INDEX1,CHAR
                GOSUB LCDWR
                NEXT
                GOSUB LCDWR2
                FOR INDEX1 = 8 TO 15
                READ MSG8 + INDEX1,CHAR
                GOSUB LCDWR
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        NEXT
        AMOUNT = 0
        RETURN

SETTING:
CHAR = CLRLCD
        GOSUB LCDCMD
        FOR INDEX1 = 0 TO 7
        READ MSG12 + INDEX1,CHAR
        GOSUB LCDWR
        NEXT
        GOSUB LCDWR2
        FOR INDEX1 = 8 TO 15
        READ MSG12 + INDEX1,CHAR
        GOSUB LCDWR
        NEXT
        RETURN

CTN:
CHAR = CLRLCD
        GOSUB LCDCMD
        FOR INDEX1 = 0 TO 7
        READ MSG9 + INDEX1,CHAR
        GOSUB LCDWR
        NEXT
        GOSUB LCDWR2
        FOR INDEX1 = 8 TO 15
        READ MSG9 + INDEX1,CHAR
        GOSUB LCDWR
        NEXT
        RETURN
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
NOCOINS:
        CHAR = CLRLCD
        GOSUB LCDCMD
  FOR INDEX1 = 0 TO 7
        READ MSG10 + INDEX1,CHAR
  GOSUB LCDWR
        NEXT
        GOSUB LCDWR2
        FOR INDEX1 = 8 TO 15
        READ MSG10 + INDEX1,CHAR
        GOSUB LCDWR
        NEXT
        AMOUNT = 0
        RETURN

SETTIME:
  CHAR = CLRLCD
        GOSUB LCDCMD
        FOR INDEX1 = 0 TO 7
        READ MSG11 + INDEX1,CHAR
        GOSUB LCDWR
        NEXT
        GOSUB LCDWR2
        FOR INDEX1 = 8 TO 15
        READ MSG11 + INDEX1,CHAR
        GOSUB LCDWR
        NEXT
        RETURN
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
ADDT:
        PAUSE 50
        BUTTON 10,0,255,255,PBOT5,0,ENOU
        GOI = GOI+1
        IF GOI = 90 THEN ZERO
        GOTO NOZE

ZERO:  GOI = 0
NOZE:   DEBUG DEC ? GOI
ENOU:
        BUTTON 8,0,255,250,PBOT4,0,ADDT
        REW = GOI
        GOTO NEX2

START2:
        PAUSE 50
        GOSUB SETTIME
        FOR WIT = 0 TO 10000
SETE:
        BUTTON 14,0,255,250,PBOT5,0,NEX
        GOSUB SETTING
        GOTO ADDT
NEX:
        BUTTON 8,0,255,250,PBOT3,0,NEX3
        GOTO NEX2
  NEX3: NEXT
NEX2:
        GOSUB READY
        PAUSE 1000
START3:
        GIB = 0
        GOTO C0B
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        GIB = 1
START:
        GOI = REW
AMOUNT = 0

START1:
        INCHA = 0
BUTTON 11,0,255,250,PBOT,0,THERE
        AMOUNT = AMOUNT + 10
        INCHA=1

THERE:
        BUTTON 12,0,255,250,PBOT2,0,CHANGE
        AMOUNT = AMOUNT + 5
        INCHA = 1

CHANGE:
        BUTTON 8,0,255,250,PBOT3,0,BSTAT
        IF AMOUNT = 0 THEN STA
        OUT15 = 1
        PAUSE 500
        OUT15 = 0
        AMOUNT = AMOUNT - 5
        INCHA = 1
        GOTO BSTAT

STA:    AMOUNT = 0

BSTAT:
        BUTTON 14,0,255,250,PBOT4,0,NOPRESS
        IF AMOUNT = 0 THEN NOTIME
        CHK = 0
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        AMOUNT = AMOUNT/5
        GOI = GOI*AMOUNT
        GOTO TIMCO1

NOTIME:
        GOSUB NOCOINS
        FREQOUT 7,250,1000,5000
        PAUSE 1000
        GOTO START3

NOPRESS:
        IF INCHA = 1 THEN DIS
        GOTO START1

DIS:
        IF AMOUNT =  0 THEN C0B
        IF AMOUNT =  5 THEN C5B
        IF AMOUNT = 10 THEN C10B
        IF AMOUNT = 15 THEN C15B
        IF AMOUNT = 20 THEN C20B
        IF AMOUNT > 20 THEN EDITOR
        GOTO START1

EDITOR:
        OLD = AMOUNT - 20
        AMOUNT = 20
        REC = OLD/5
        FOR COUN = 1 TO REC
        OUT15 = 1
        PAUSE 500
        OUT15 = 0
        PAUSE 500
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            NEXT
            GOTO C20B

TIMCO1:GOSUB CTT

TIMCO:
            TIH = GOI
            FOR AUY = 1 TO TIH
            GOSUB DELAY
            IF CHK = 1 THEN NOADD
            NEXT
            GOSUB CTN
            FREQOUT 7,500,1000,5000
            PAUSE 200
            FREQOUT 7,500,1000,5000
            PAUSE 200
            FREQOUT 7,1000,1000,5000
            FOR ADD = 1 TO 5000
            BUTTON 11,0,255,250,PBOT2,0,NO1
            GOI = REW
            GOI = GOI*2
            GOTO TIMCO1

NO1:
            BUTTON 12,0,255,250,PBOT2,0,NO
            GOI = REW
            GOTO TIMCO1

NO:         NEXT
  NOADD:
            PAUSE 1000
            OUT6 = 0
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

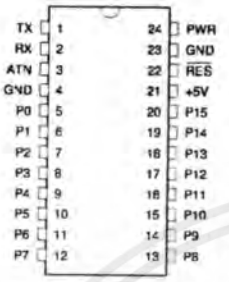
```
        GOTO START3

LCDCMD: LOW RS

LCDWR:
        OUTA = CHAR.HIGHNIB
        PULSOUT E,1
        OUTA = CHAR.LOWNIB
        PULSOUT E,1
        HIGH RS
        RETURN
LCDWR2:
        CHAR2 = %11000000
        LOW RS
        OUTA = CHAR2.HIGHNIB
        PULSOUT E,1
        OUTA = CHAR2.LOWNIB
        PULSOUT E,1
        HIGH RS
        RETURN

DELAY:
        OUT6 = 1
        FOR DEL = 0 TO 36100
        BUTTON 8,0,255,250,PBOT4,0,DE1
        CHK = 1
        RETURN
DE1:
        NEXT
        RETURN
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ค.

DATA SHEET

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Pin	Name	Description	Comments
1	TX	Serial output	Connect to pin 2 of PC serial DB9 (RX) *
2	RX	Serial input	Connect to pin 3 of PC serial DB9 (TX) *
3	ATN	Active-high reset	Connect to pin 4 of PC serial DB9 (DTR) *
4	GND	Serial ground	Connect to pin 5 of PC serial DB9 (GND) *
5	P0	I/O pin 0	Each pin can source 20 ma and sink 25 ma.
6	P1	I/O pin 1	
7	P2	I/O pin 2	P0-P7 and P8-P15, as groups, can each
8	P3	I/O pin 3	source a total of 40 ma and sink 50 ma.
9	P4	I/O pin 4	
10	P5	I/O pin 5	
11	P6	I/O pin 6	
12	P7	I/O pin 7	
13	P8	I/O pin 8	
14	P9	I/O pin 9	
15	P10	I/O pin 10	
16	P11	I/O pin 11	
17	P12	I/O pin 12	
18	P13	I/O pin 13	
19	P14	I/O pin 14	
20	P15	I/O pin 15	
21	+5V **	+5V supply	5-volt input or regulated output.
22	RES	Active-low reset	Pull low to reset; goes low during reset.
23	GND	System ground	
24	PWR **	Regulator input	Voltage regulator input; takes 5-15 volts.

* For automatic serial port selection by the BASIC Stamp II software, there must also be a connection from DSR (DB9 pin 6) to RTS (DB9 pin 7). This connection is made on the BASIC Stamp II carrier board. If you are not using the carrier board, then you must make this connection yourself, or use the command-line option to tell the software which serial port to use.

** During normal operation, the BASIC Stamp II takes about 7 mA. In various power-down modes, consumption can be reduced to about 50 µA.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

The following list is a summary of the PBASIC instructions used by the BASIC Stamp II.

◆ This symbol indicates new or greatly improved instructions (compared to the BASIC Stamp I).

BRANCHING

IF...THEN	Compare and conditionally branch.
BRANCH	Branch to address specified by offset.
GOTO	Branch to address.
GOSUB	Branch to subroutine at address. GOSUBs may be nested up to four levels deep, and you may have up to 255 GOSUBs in your program.
RETURN	Return from subroutine.

LOOPING

FOR...NEXT	Establish a FOR-NEXT loop.

NUMERICS

LOOKUP	Lookup data specified by offset and store in variable. This instruction provides a means to make a lookup table.
LOOKDOWN	Find target's match number (0-N) and store in variable.
RANDOM	Generate a pseudo-random number.

DIGITAL I/O

INPUT	Make pin an input
OUTPUT	Make pin an output.
REVERSE	If pin is an output, make it an input. If pin is an input, make it an output.
LOW	Make pin output low.
HIGH	Make pin output high.
TOGGLE	Make pin an output and toggle state.
PULSIN	Measure an input pulse (resolution of 2 µs).

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	PULSOUT	Output a timed pulse by inverting a pin for some time (resolution of 2 µs).
	BUTTON	Debounce button, perform auto-repeat, and branch to address if button is in target state.
◆	SHIFTIN	Shift bits in from parallel-to-serial shift register.
◆	SHIFTOUT	Shift bits out to serial-to-parallel shift register.
◆	COUNT	Count cycles on a pin for a given amount of time (0 - 125 kHz, assuming a 50/50 duty cycle).
◆	XOUT	Generate X-10 powerline control codes. For use with TW523 or TW513 powerline interface module.

SERIAL I/O

◆	SERIN	Serial input with optional qualifiers, time-out, and flow control. If qualifiers are given, then the instruction will wait until they are received before filling variables or continuing to the next instruction. If a time-out value is given, then the instruction will abort after receiving nothing for a given amount of time. Baud rates of 300 - 50,000 are possible (0 - 19,200 with flow control). Data received must be N81 (no parity, 8 data bits, 1 stop bit) or E71 (even parity, 7 data bits, 1 stop bit).
◆	SEROUT	Send data serially with optional byte pacing and flow control. If a pace value is given, then the instruction will insert a specified delay between each byte sent (pacing is not available with flow control). Baud rates of 300 - 50,000 are possible (0 - 19,200 with flow control). Data is sent as N81 (no parity, 8 data bits, 1 stop bit) or E71 (even parity, 7 data bits, 1 stop bit).

ANALOG I/O

	PWM	Output PWM, then return pin to input. This can be used to output analog voltages (0-5V) using a capacitor and resistor.
◆	RCTIME	Measure an RC charge/discharge time. Can be used to measure potentiometers.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SOUND

◆ FREQOUT	Generate one or two sinewaves of specified frequencies (each from 0 - 32767 hz.).
◆ DTMFOUT	Generate DTMF telephone tones.

EEPROM ACCESS

◆ DATA	Store data in EEPROM before downloading PBASIC program.
READ	Read EEPROM byte into variable.
WRITE	Write byte into EEPROM.

TIME

PAUSE	Pause execution for 0–65535 milliseconds.

POWER CONTROL

NAP	Nap for a short period. Power consumption is reduced.
SLEEP	Sleep for 1-65535 seconds. Power consumption is reduced to approximately 50 µA.
END	Sleep until the power cycles or the PC connects. Power consumption is reduced to approximately 50 µA.

PROGRAM DEBUGGING

DEBUG	Send variables to PC for viewing.

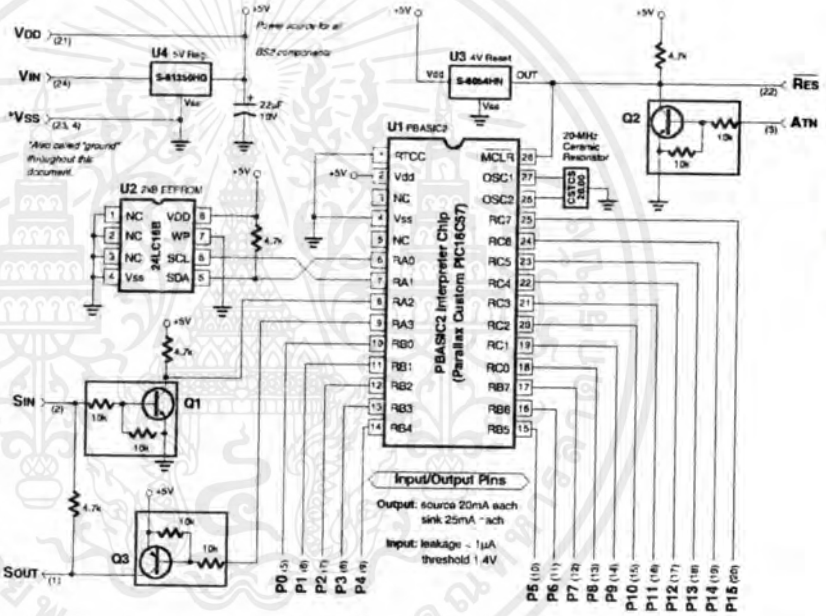
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

BS2 Hardware

Figure H-1 is a schematic diagram of the BASIC Stamp II (BS2). In this section we'll describe each of the major components and explain its function in the circuit.

Figure H-1

Schematic Diagram of the BASIC Stamp II (BS2-IC rev. A)



NOTES
1. This diagram depicts the DIP/SOIC version of the PBASIC2 interpreter chip, since users wishing to construct a BS2 from discrete components are most likely to use those parts. Contact Parallax for a schematic depicting the SSOP (ultra-small surface mount) package used in the BS2-IC module.

2. Numbers in parentheses—(#)—are pin numbers on the BS2-IC module. The BS2-IC has the form factor of a 24-pin, 0.6" DIP.

3. Q1, Q2 and Q3 are Rohm part numbers. Other components may be substituted in custom circuits, subject to appropriate design. Contact Parallax for design assistance.

4. U3 and U4 are Seiko part numbers. Other components may be substituted in custom circuits, subject to appropriate design. Contact Parallax for design assistance.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PBASIC2 Interpreter Chip (U1)

The brain of the BS2 is a custom PIC16C57 microcontroller (U1). U1 is permanently programmed with the PBASIC2 instruction set. When you program the BS2, you are telling U1 to store symbols, called *tokens*, in EEPROM memory (U2). When your program runs, U1 retrieves tokens from memory (U2), interprets them as PBASIC2 instructions, and carries out those instructions.

U1 executes its internal program at 5 million instructions per second. Many internal instructions go into a single PBASIC2 instruction, so PBASIC2 executes more slowly—approximately 3000 to 4000 instructions per second.

The PIC16C57 controller has 20 input/output (I/O) pins; in the BS2 circuit, 16 of these are available for general use by your programs. Two others may also be used for serial communication. The remaining two are used solely for interfacing with the EEPROM and may not be used for anything else.

The general-purpose I/O pins, P0 through P15, can interface with all modern 5-volt logic, from TTL (transistor-transistor logic) through CMOS (complementary metal-oxide semiconductor). To get technical, their properties are very similar to those of 74HCTxxx-series logic devices.

The direction—input or output—of a given pin is entirely under the control of your program. When a pin is an input, it has very little effect on circuits connected to it, with less than 1 microampere (µA) of current leaking in or out. You may be familiar with other terms for input mode like *tristate, high-impedance*, or *hi-Z*.

There are two purposes for putting a pin into input mode: (1) To passively read the state (1 or 0) of the pin as set by external circuitry, or (2) To disconnect the output drivers from the pin. For lowest current draw, inputs should always be as close to +5V or ground as possible. They should not be allowed to float. Unused pins that are not connected to circuitry should be set to output.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

When a pin is an output, it is internally connected to ground or +5V through a very efficient CMOS switch. If it is lightly loaded (< 1mA), the output voltage will be within a few millivolts of the power supply rail (ground for 0; +5V for 1). Pins can sink as much as 25mA (outputting 0) and source up to 20 mA (outputting 1). Each of the two eight-pin ports should not carry more than a total of 50mA (sink) or 40mA (source). Pins P0 through P7 make up one port; P8 through P15 the other.

2048-byte Erasable Memory Chip (U2)

U1 is permanently programmed at the factory and cannot be reprogrammed, so your PBASIC2 programs must be stored elsewhere. That's the purpose of U2, the 24LC16B electrically erasable, programmable read-only memory (EEPROM). EEPROM is a good medium for program storage because it retains data without power, but can be reprogrammed easily.

EEPROMs have two limitations: (1) They take a relatively long time (as much as several milliseconds) to write data into memory, and (2) There is a limit to the number of writes (approximately 10 million) they will accept before wearing out. Because the primary purpose of the BS2's EEPROM is program storage, neither of these is normally a problem. It would take many lifetimes to write and download 10 million PBASIC2 programs! However, when you use the PBASIC2 Write instruction to store data in EEPROM space be sure to bear these limitations in mind.

Reset Circuit (U3)

When you first power up the BS2, it takes a fraction of a second for the supply to reach operating voltage. During operation, weak batteries, varying input voltages or heavy loads may cause the supply voltage to wander out of acceptable operating range. When this happens, normally infallible processor and memory chips (U1 and U2) can make mistakes or lock up. To prevent this, U1 must be stopped and reset until the supply stabilizes. That is the job of U3, the S-8045HN reset circuit. When the supply voltage is below 4V, U3 puts a logic low on U1's master-clear reset (MCLR) input. This stops U1 and causes all of its I/O lines to electrically disconnect. In reset, U1 is dormant; alive but inert.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

When the supply voltage is above 4V, U3 allows its output to be pulled high by a 4.7k resistor to +5V, which also puts a high on U1's MCLR input. U1 starts its internal program at the beginning, which in turn starts your PBASIC2 program from the beginning.

Power Supply (U4)

The previous discussion of the reset circuit should give you some idea of how important a stable power supply is to correct operation of the BS2. The first line of defense against power-supply problems is U4, the S-81350HG 5-volt regulator. This device accepts a range of slightly over 5V up to 15V and regulates it to a steady 5V. This regulator draws minimal current for its own use, so when your program tells the BS2 to go into low-power Sleep, End or Nap modes, the total current draw averages out to approximately 100 microamperes (µA). (That figure assumes no loads are being driven and that all I/O pins are at ground or +5V.) When the BS2 is active, it draws approximately 8mA. Since U4 can provide up to 50mA, the majority of its capacity is available for powering your custom circuitry.

Circuits requiring more current than U4 can provide may incorporate their own 5V supply. Connect this supply to VDD and leave U4's input (VIN) open.

Note that figure H-1 uses CMOS terms for the power supply rails, VDD for the positive supply and VSS for ground or 0V reference. These terms are correct because the main components are CMOS. Don't be concerned that other circuits you may come across use different nomenclature; for our purposes, the terms VDD, VCC, and +5V are interchangeable, as are VSS, earth (British usage) and ground.

Serial Host Interface (Q1, Q2, and Q3)

The BS2 has no keyboard or monitor, so it relies on PC-based host software to allow you to write, edit, download and debug PBASIC2 programs. The PC communicates with the BS2 through an RS-232 (COM port) interface consisting of pins SIN, SOUT, and ATN (serial in, serial out, and attention, respectively).

RS-232 uses two signaling voltages to represent the logic states 0 and 1; +12V is 0 and –12V is 1. When an RS-232 sender has nothing to say, it

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

leaves its output in the 1 state (-12V). To begin a transmission, it outputs a 0 (+12V) for one bit time (the baud rate divided into 1 second; e.g., bit time for 2400 baud = 1/2400 = 416.6µs).

You can see how the BS2 takes advantage of these characteristics in the design of its serial interface. NPN transistor Q1 serves as a serial line receiver. When SIN is negative, Q1 is switched off, so the 4.7k resistor on its collector puts a high on pin RA2 of U1, the PBASIC2 interpreter chip. When SIN goes high, Q1 switches on, putting a 0 on RA2/U1.

SOUT transmits data from U1 to the PC. When SOUT outputs a 1, it borrows the negative resting-state voltage of SIN and reflects it back to SOUT through a 4.7k resistor. When SOUT transmits a 0, it turns on PNP transistor Q3 to put a +5V level on SOUT. In this way the BS2 outputs +5/–12V RS-232.

Of course, this method works only with the cooperation of the PC software, which must not transmit serial data at the same time the BS2 is transmitting.

The ATN line interfaces with the data-terminal ready (DTR) handshaking line of the PC COM port. Electrically, it works like the SIN line receiver, with a +12V signal at ATN turning on the Q2 transistor, pulling its collector to ground. Q2's collector is connected to the MCLR (reset) line of the PBASIC2 interpreter chip, so turning on Q2 resets U1. During programming, the STAMP2 host program pulses ATN high to reset U1, then transmits a signal to U1 through SIN indicating that it wants to download a new program. Other than when it wants to initiate programming, the STAMP2 host program holds ATN at –12V, allowing U1 to run normally.

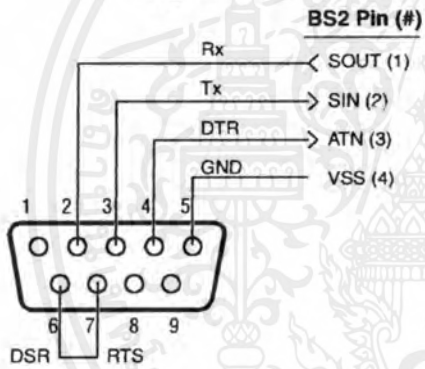
Your PBASIC2 programs may use the serial host interface to communicate with PC programs other than the STAMP2 host program. The only requirement is that ATN must be either disconnected or at less than +1V to avoid unintentionally resetting the BS2. See the Serin listing for further information.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PC-to-BS2 Connector Hookup

Figure H-2 shows how a DB9 programming connector for the BS2 is wired. This connector allows the PC to reset the BS2 for programming, download programs, and receive Debug data from the BS2. An additional pair of connections, pins 6 and 7 of the DB9 socket, lets the STAMP2 host software identify the port to which the BS2 is connected. If you plan to construct your own carrier board or make temporary programming connections to a BS2 on a prototyping board, use this drawing as a guide. If you also want to use this host interface connection to communicate between the BS2 and other PC programs, see the writeup in the Serin listing for suggestions.

Figure H-2



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

BS2 Memory Organization

The BS2 has two kinds of memory; RAM for variables used by your program, and EEPROM for storing the program itself. EEPROM may also be used to store long-term data in much the same way that desktop computers use a hard drive to hold both programs and files.

An important distinction between RAM and EEPROM is this:

- RAM loses its contents when the BS2 loses power; when power returns, all RAM locations are cleared to 0s.
- EEPROM retains the contents of memory, with or without power, until it is overwritten (such as during the program-downloading process or with a Write instruction.)

In this section, we'll look at both kinds of BS2 memory, how it's organized, and how to use it effectively. Let's start with RAM.

BS2 Data Memory (RAM)

The BS2 has 32 bytes of RAM. Of these, 6 bytes are reserved for input, output, and direction control of the 16 input/output (I/O) pins. The remaining 26 bytes are available for use as variables.

The table below is a map of the BS2's RAM showing the built-in PBASIC names.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Table M-1. BS2 Memory Map

Stamp II I/O and Variable Space				
Word Name	**Byte Name**	**Nibble Names**	**Bit Names**	**Special Notes**
INS	INL INH	INA, INB, INC, IND	IN0 - IN7, IN8 - IN15	Input pins; word, byte, nibble and bit addressable.
OUTS	OUTL OUTH	OUTA, OUTB, OUTC, OUTD	OUT0 - OUT7, OUT8 - OUT15	Output pins; word, byte, nibble and bit addressable.
DIRS	DIRL DIRH	DIRA, DIRB, DIRC, DIRD	DIR0 - DIR7, DIR8 - DIR15	I/O pin direction control; word, byte, nibble and bit addressable.
W0	B0 B1			General Purpose; word, byte, nibble and bit addressable.
W1	B2 B3			General Purpose; word, byte, nibble and bit addressable.
W2	B4 B5			General Purpose; word, byte, nibble and bit addressable.
W3	B6 B7			General Purpose; word, byte, nibble and bit addressable.
W4	B8 B9			General Purpose; word, byte, nibble and bit addressable.
W5	B10 B11			General Purpose; word, byte, nibble and bit addressable.
W6	B12 B13			General Purpose; word, byte, nibble and bit addressable.
W7	B14 B15			General Purpose; word, byte, nibble and bit addressable.
W8	B16 B17			General Purpose; word, byte, nibble and bit addressable.
W9	B18 B19			General Purpose; word, byte, nibble and bit addressable.
W10	B20 B21			General Purpose; word, byte, nibble and bit addressable.
W11	B22 B23			General Purpose; word, byte, nibble and bit addressable.
W12	B24 B25			General Purpose; word, byte, nibble and bit addressable.

The Input/Output (I/O) Variables

As the map shows, the first three words of the memory map are associated with the Stamp's 16 I/O pins. The word variable INS is unique in that it is read-only. The 16 bits of INS reflect the bits present at Stamp I/O pins P0 through P15. It may only be read, not written. OUTS con-

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

tains the states of the 16 output latches. DIRS controls the direction (input or output) of each of the 16 pins.

If you are new to devices that can change individual pins between input and output, the INS/OUTS/DIRS trio may be a little confusing, so we'll walk through the possibilities.

A 0 in a particular DIRS bit makes the corresponding pin, P0 through P15, an input. So if bit 5 of DIRS is 0, then P5 is an input. A pin that is an input is at the mercy of circuitry outside the Stamp; the Stamp cannot change its state. When the Stamp is first powered up, all memory locations are cleared to 0, so all pins are inputs (DIRS = %0000000000000000).

A 1 in a DIRS bit makes the corresponding pin an output. This means that the corresponding bit of OUTS determines that pin's state.

Suppose all pins' DIRS are set to output (1s) and you look at the contents of INS. What do you see? You see whatever is stored in the variable OUTS.

OK, suppose all pins' DIRS are set to input (0s) and external circuits connected to the pins have them all seeing 0s. What happens to INS if you write 1s to all the bits of OUTS? Nothing. INS will still contain 0s, because with all pins set to input, the external circuitry is in charge. However, when you change DIRS to output (1s), the bits stored in OUTS will appear on the I/O pins.

These possibilities are summarized in the Figure M-1 below. To avoid making the table huge, we'll look at only one bit. The rules shown for a single bit apply to all of the I/O bits/pins. Additionally, the external circuitry producing the "external state" listed in the table can be overridden by a Stamp output. For example, a 10k resistor to +5V will place a 1 on an input pin, but if that pin is changed to output and cleared to 0, a 0 will appear on the pin, just as the table shows. However, if the pin is connected directly to +5V and changed to output 0, the pin's state will remain 1. The Stamp simply cannot overcome a direct short, and will probably be damaged in the bargain.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Figure M-1. Interaction of DIRS, INS and OUTS

The DIRS register controls which I/O pins are inputs and which are outputs. When set to input (0), the corresponding bit in the OUTS register is disconnected and ignored.

When set to output (1), the corresponding bit in the OUTS register is connected.

NOTE: "X" indicates state could be a 1 or a 0 and does not affect other elements.
"?" indicates state is unknown and could change erratically.

To summarize: DIRS determines whether a pin's state is set by external circuitry (input, 0) or by the state of OUTS (output, 1). INS always matches the actual states of the I/O pins, whether they are inputs or outputs. OUTS holds bits that will only appear on pins whose DIRS bits are set to output.

In programming the BS2, it's often more convenient to deal with individual bytes, nibbles or bits of INS, OUTS and DIRS rather than the entire 16-bit words. PBASIC2 has built-in names for these elements, listed below. When we talk about the low byte of these words, we mean the byte corresponding to pins P0 through P7.

Table M-2. Predefined Names for Elements of DIRS, INS and OUTS

DIRS	INS	OUTS	The entire 16-bit word
DIRL	INL	OUTL	The low byte of the word
DIRH	INH	OUTH	The high byte of the word
DIRA	INA	OUTA	The low nibble of low byte
DIRB	INB	OUTB	The high nibble of low byte
DIRC	INC	OUTC	The low nibble of high byte
DIRD	IND	OUTD	The high nibble of high byte
DIR0	IN0	OUT0	The low bit; corresponds to P0
	...(continues 1 through 14)...		Bits 1 - 14; corresponds to P1 through P14
DIR15	IN15	OUT15	The high bit; corresponds to P15

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Table M-3 lists all the variable modifiers. PBASIC2 lets you apply these modifiers to any variable name, including fixed variables and I/O variables, and to combine them in any fashion that makes sense. For example, it will allow:

```
rhino       var     word                    ' A 16-bit variable.
eye         var     rhino.highbyte.lownib.bit1    ' A bit.
```

Table M-3. Variable Modifiers

SYMBOL	DEFINITION
LOWBYTE	'low byte of a word
HIGHBYTE	'high byte of a word
BYTE0	'byte 0 (low byte) of a word
BYTE1	'byte 1 (high byte) of a word
LOWNIB	'low nibble of a word or byte
HIGHNIB	'high nibble of a word or byte
NIB0	'nib 0 of a word or byte
NIB1	'nib 1 of a word or byte
NIB2	'nib 2 of a word
NIB3	'nib 3 of a word
LOWBIT	'low bit of a word, byte, or nibble
HIGHBIT	'high bit of a word, byte, or nibble
BIT0	'bit 0 of a word, byte, or nibble
BIT1	'bit 1 of a word, byte, or nibble
BIT2	'bit 2 of a word, byte, or nibble
BIT3	'bit 3 of a word, byte, or nibble
BIT4	'bit 4 of a word or byte
BIT5	'bit 5 of a word or byte
BIT6	'bit 6 of a word or byte
BIT7	'bit 7 of a word or byte
BIT8	'bit 8 of a word
BIT9	'bit 9 of a word
BIT10	'bit 10 of a word
BIT11	'bit 11 of a word
BIT12	'bit 12 of a word
BIT13	'bit13 of a word
BIT14	'bit14 of a word
BIT15	'bit15 of a word

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Binary (two-argument) Operators

Sixteen Binary Operators are listed and explaned below.

Table M-6. Binary Operators

Operator	Description
+	Addition
-	Subtraction
/	Division
//	Remainder of division
*	Multiplication
**	High 16-bits of multiplication
*/	Multiply by 8-bit whole and 8-bit part
MIN	Limits a value to specified low
MAX	Limits a value to specified high
DIG	Returns specified digit of number
<<	Shift bits left by specified amount
>>	Shift bits right by specified amount
REV	Reverse specified number of bits
&	Bitwise AND of two values
|	Bitwise OR of two values
^	Bitwise XOR of two values

+

Adds variables and/or constants, returning a 16-bit result. Works exactly as you would expect with unsigned integers from 0 to 65535. If the result of addition is larger than 65535 the carry bit will be lost. If the values added are signed 16-bit numbers and the destination is a 16-bit variable, the result of the addition will be correct in both sign and value. For example, the expression -1575 + 976 will result in the signed value -599. See for yourself:

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Table I-1. Debug Modifiers

Modifier	Effect	Notes
ASC?	Displays "variablename = 'character'" + carriage return; where character is an ASCII character.	1
DEC {1..5}	Decimal text, optionally fixed for 1 to 5 digits	
SDEC {1..5}	Signed decimal text, optionally fixed for 1 to 5 digits	1, 2
HEX {1..4}	Hexadecimal text, optionally fixed for 1 to 4 digits	1
SHEX {1..4}	Signed hex text, optionally fixed for 1 to 4 digits	1, 2
IHEX {1..4}	Indicated hex text ($ prefix; e.g., $7A3), optionally fixed for 1 to 4 digits	1
ISHEX {1..4}	Indicated signed hex text, optionally fixed for 1 to 4 digits	1, 2
BIN {1..16}	Binary text, optionally fixed for 1 to 16 digits	1
SBIN {1..16}	Signed binary text, optionally fixed for 1 to 16 digits	1, 2
IBIN {1..16}	Indicated binary text (% prefix; e.g., %10101100), optionally fixed for 1 to 16 digits	1, 2
ISBIN {1..16}	Indicated signed binary text, optionally fixed for 1 to 16 digits	1, 2
STR bytearray	Display an ASCII string from bytearray until byte = 0.	
STR bytearray\n	Display an ASCII string consisting of n bytes from bytearray.	
REP byte\n	Display an ASCII string consisting of byte repeated n times (e.g., REP "X"\10 sends XXXXXXXXXX).	

NOTES:

(1) Fixed-digit modifiers like DEC4 will pad text with leading 0s if necessary; e.g., DEC4 65 sends 0065. If a number is larger than the specified number of digits, the leading digits will be dropped; e.g., DEC4 56422 sends 6422.

(2) Signed modifiers work under two's complement rules, same as PBASIC2 math. Value must be no less than a word variable in size.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

BASIC Stamp II Application Notes

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ASCII Chart

Control Codes			Printing Characters					
Name/Function	***Char**	**Code**	**Char**	**Code**	**Char**	**Code**	**Char**	**Code**
null	NUL	0	<space>	32	@	64	`	96
start of heading	SOH	1	!	33	A	65	a	97
start of text	STX	2	"	34	B	66	b	98
end of text	ETX	3	#	35	C	67	c	99
end of xmit	EOT	4	$	36	D	68	d	100
enquiry	ENQ	5	%	37	E	69	e	101
acknowledge	ACK	6	&	38	F	70	f	102
bell	BEL	7	'	39	G	71	g	103
backspace	BS	8	(	40	H	72	h	104
horizontal tab	HT	9	)	41	I	73	i	105
line feed	LF	10	*	42	J	74	j	106
vertical tab	VT	11	+	43	K	75	k	107
form feed	FF	12	,	44	L	76	l	108
carriage return	CR	13	-	45	M	77	m	109
shift out	SO	14	.	46	N	78	n	110
shift in	SI	15	/	47	O	79	o	111
data line escape	DLE	16	0	48	P	80	p	112
device control 1	DC1	17	1	49	Q	81	q	113
device control 2	DC2	18	2	50	R	82	r	114
device control 3	DC3	19	3	51	S	83	s	115
device control 4	DC4	20	4	52	T	84	t	116
non acknowledge	NAK	21	5	53	U	85	u	117
synchronous idle	SYN	22	6	54	V	86	v	118
end of xmit block	ETB	23	7	55	W	87	w	119
cancel	CAN	24	8	56	X	88	x	120
end of medium	EM	25	9	57	Y	39	y	121
substitute	SUB	26	:	58	Z	90	z	123
escape	ESC	27	;	59	[	91	{	124
file separator	FS	28	<	60	\	92	|	125
group separator	GS	29	=	61	]	93	}	126
record separator	RS	30	>	62	^	94	~	127
unit separator	US	31	?	63	_	95	<delete>	128

* Note that the control codes have no standardized screen symbols. The characters listed for them are just names used in referring to these codes. For example, to move the cursor to the beginning of the next line of a pritner or terminal often requires sending linefeed and carriage return codes. This common pair is referred to as "LF/CR."

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Appendix A

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

The following table shows the reserved words for each stamp module.

BASIC STAMP I		BASIC STAMP II		
AND	ON2400	ABS	HOME	OUTL
B0..B13	OR	AND	IHEX	OUTPUT
BIT0..BIT15	OT300	ASC	IHEX1..IHEX4	OUTS
BRANCH	OT600	BELL	IF	PAUSE
BSAVE	OT1200	BKSP	IN0..IN15	RCTIME
BUTTON	OT2400	BIN	INA	REV
DEBUG	OUTPUT	BIN1..BIN4	INB	PULSIN
DIR0..DIR7	PAUSE	BIT	INC	PULSOUT
DIRS	PIN0..PIN7	BIT0..BIT15	IND	PWM
EEPROM	PINS	BRANCH	INH	RANDOM
END	PORT	BRIGHT	INL	READ
FOR	POT	BUTTON	INPUT	REP
GOSUB	PULSIN	BYTE	INS	REVERSE
GOTO	PULSOUT	CLS	ISBIN	SBIN
HIGH	PWM	CON	ISBIN1..ISBIN16	SBIN1..SBIN16
IF	RANDOM	COS	ISHEX	SDEC
INPUT	READ	COUNT	ISHEX1..ISHEX4	SDEC1..SDEC5
LET	REVERSE	CR	LIGHTSON	SERIN
LOOKDOWN	SERIN	DATA	LOOKDOWN	SEROUT
LOOKUP	SEROUT	DCD	LOOKUP	SHEX
LOW	SLEEP	DEBUG	LOW	SHEX1..SHEX4
MAX	SOUND	DEC	LOWBIT	SHIFTIN
MIN	STEP	DEC1..DEC5	LOWNIB	SHIFTOUT
N300	SYMBOL	DIG	LSBFIRST	SIN
N600	T300	DIM	LSBPOST	SKIP
N1200	T600	DIR0..DIR15	LSBPRE	SLEEP
N2400	T1200	DIRA	MAX	STEP
NAP	T2400	DIRB	MIN	STOP
NEXT	THEN	DIRC	MSBFIRST	STR
ON300	TOGGLE	DIRD	MSBPOST	SQR
ON600	W0..W6	DIRH	MSBPRE	TAB
ON1200	WRITE	DIRL	NAP	THEN
		DIRS	NCD	TO
		DTMFOUT	NEXT	TOGGLE
		END	NIB	UNITOFF
		FOR	NIB0..NIB3	UNITON
		FREQOUT	NOT	UNITSOFF
		GOSUB	OR	VAR
		GOTO	OUT0..OUT15	WAIT
		HEX	OUTA	WAITSTR
		HEX1..HEX4	OUTB	WORD
		HIGH	OUTC	WRITE
		HIGHBIT	OUTD	XOR
		HIGHNIB	OUTH	XOUT

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Appendix B

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- CONSTANTVALUE is a valid number in decimal, hexidecimal, binary or ascii.

Stamp I I/O and Variable Space			
Word Name	**Byte Name**	**Bit Names**	**Special Notes**
PORT	PINS	PIN0 - PIN7	I/O pins; bit addressable.
	DIRS	DIR0 - DIR7	I/O pin direction control; bit addressable.
W0	B0	BIT0 - BIT7	Bit addressable.
	B1	BIT8 - BIT15	Bit addressable.
W1	B2		
	B3		
W2	B4		
	B5		
W3	B6		
	B7		
W4	B8		
	B9		
W5	B10		
	B11		
W6	B12		Used by GOSUB instruction.
	B13		Used by GOSUB instruction.

BASIC Stamp II

The RAM space of the BASIC Stamp II consists of sixteen words of 16 bits each. Each word and each byte within the word has a unique, predefined name similar to the Stamp I and shown in the table below.

The first three words, named INS, OUTS and DIRS, are reserved to allow access and control over the 16 I/O pins on the Stamp II. These reserved words represent the input states, output states and directions of the pins respectively and are the Stamp II version of the single control word, PORT, on the Stamp I. In comparison to the Stamp I, the control registers' size has been doubled and the I/O register PINS has been split into two words, INS and OUTS, for flexibility. Each word consists of a predefined name for its byte, nibble and bit parts.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- CONSTANTNAME is a series of characters (letters, numbers and underscores but not starting with a number) up to 32 characters in length.
- CONSTANTEXPRESSION is a numerical expression in decimal, hexidecimal, binary or ascii using only numbers and the +, -, *, /, &, |, ^, << or >> operators. NOTE: Parentheses are not allowed and expressions are always computed using 16-bits.

Stamp II I/O and Variable Space

Word Name	Byte Name	Nibble Names	Bit Names	Special Notes
INS	INL INH	INA, INB, INC, IND	IN0 - IN7, IN8 - IN15	Input pins; word, byte, nibble and bit addressable.
OUTS	OUTL OUTH	OUTA, OUTB, OUTC, OUTD	OUT0 - OUT7, OUT8 - OUT15	Output pins; word, byte, nibble and bit addressable.
DIRS	DIRL DIRH	DIRA, DIRB, DIRC, DIRD	DIR0 - DIR7, DIR8 - DIR15	I/O pin direction control; word byte, nibble and bit addressable.
W0	B0 B1			General Purpose; word, byte, nibble and bit addressable.
W1	B2 B3			General Purpose; word, byte, nibble and bit addressable.
W2	B4 B5			General Purpose; word, byte, nibble and bit addressable.
W3	B6 B7			General Purpose; word, byte, nibble and bit addressable.
W4	B8 B9			General Purpose; word, byte, nibble and bit addressable.
W5	B10 B11			General Purpose; word, byte, nibble and bit addressable.
W6	B12 B13			General Purpose; word, byte, nibble and bit addressable.
W7	B14 B15			General Purpose; word, byte, nibble and bit addressable.
W8	B16 B17			General Purpose; word, byte, nibble and bit addressable.
W9	B18 B19			General Purpose; word, byte, nibble and bit addressable.
W10	B20 B21			General Purpose; word, byte, nibble and bit addressable.
W11	B22 B23			General Purpose; word, byte, nibble and bit addressable.
W12	B24 B25			General Purpose; word, byte, nibble and bit addressable.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Symbol Conversion: BS1 R BS2

1. Remove the 'SYMBOL' directive from variable or constant declarations.
2. On all variable declarations, replace the predefined register name, to the right of the '=', with the corresponding variable type or register name according to the following table:

BS1 to BS2 Register Allocation Conversion	
Stamp I Register Name	**Stamp II Variable Type / Register Name**
PORT	NO EQUIVALENT*
PINS or PIN0..PIN7	INS / OUTS or IN0..IN7 / OUT0..OUT7**
DIRS or DIR0..DIR7	DIRS or DIR0..DIR7
W0..W6	WORD
B0..B13	BYTE
BIT0..BIT15	BIT

* The PORT control register has been split into three registers, INS, OUTS and DIRS, on the Stamp II. There is no predefined name representing all registers as a group as in the Stamp I. Additional symbol and / or program structure and logic changes are necessary to access all three registers properly.

** The Stamp I PINS register has been split into two registers, INS and OUTS, in the Stamp II. Each register now has a specific task, input or output, rather than a dual task, both input and output, as in the Stamp I. If the Stamp I program used the symbol assigned to PINS for both input and output, an additional symbol is necessary to access both functions. This may also require further changes in program structure and logic.

1. On all variable declarations, replace the equal sign, '=', with 'VAR'.
2. On all constant declarations, replace the equal sign, '=', with 'CON'.

Variable or Constant Conversion: BS1 Q BS2

1. Insert the 'SYMBOL' directive before the variable's name or constant's name in the declaration.
2. On all variable declarations, replace the variable type or register name, to the right of the '=', with the corresponding, predefined register name according to the following table:

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

BS2 to BS1 Register Allocation Conversion	
Stamp II Variable Type / Register Name	**Stamp I Register Name**
INS	PINS
OUTS	PINS
DIRS	DIRS
WORD	W0..W6
BYTE	B0..B13
NIB	B0..B13*
BIT	BIT0..BIT15**

* There are no registers on the Stamp I which are nibble addressable. The best possible solution is to place one or two nibble variables within a byte register and modify the code accordingly.

** The only general purpose registers on the Stamp I which are bit addressable are B0 and B1. BIT0..BIT7 correspond to the bits within B0 and BIT8..BIT15 correspond to the bits within B1. If you have a set of bit registers in the Stamp II program, you should reserve B0 and B1 for this bit usage; i.e.: do not assign any other symbols to B0 or B1.

3. On all variable and constant declarations, replace the variable or constant directive, 'VAR' or 'CON', with an equal sign, '='.

Assignment Conversion: BS1 Q BS2

1 Remove the 'LET' command if it is specified.

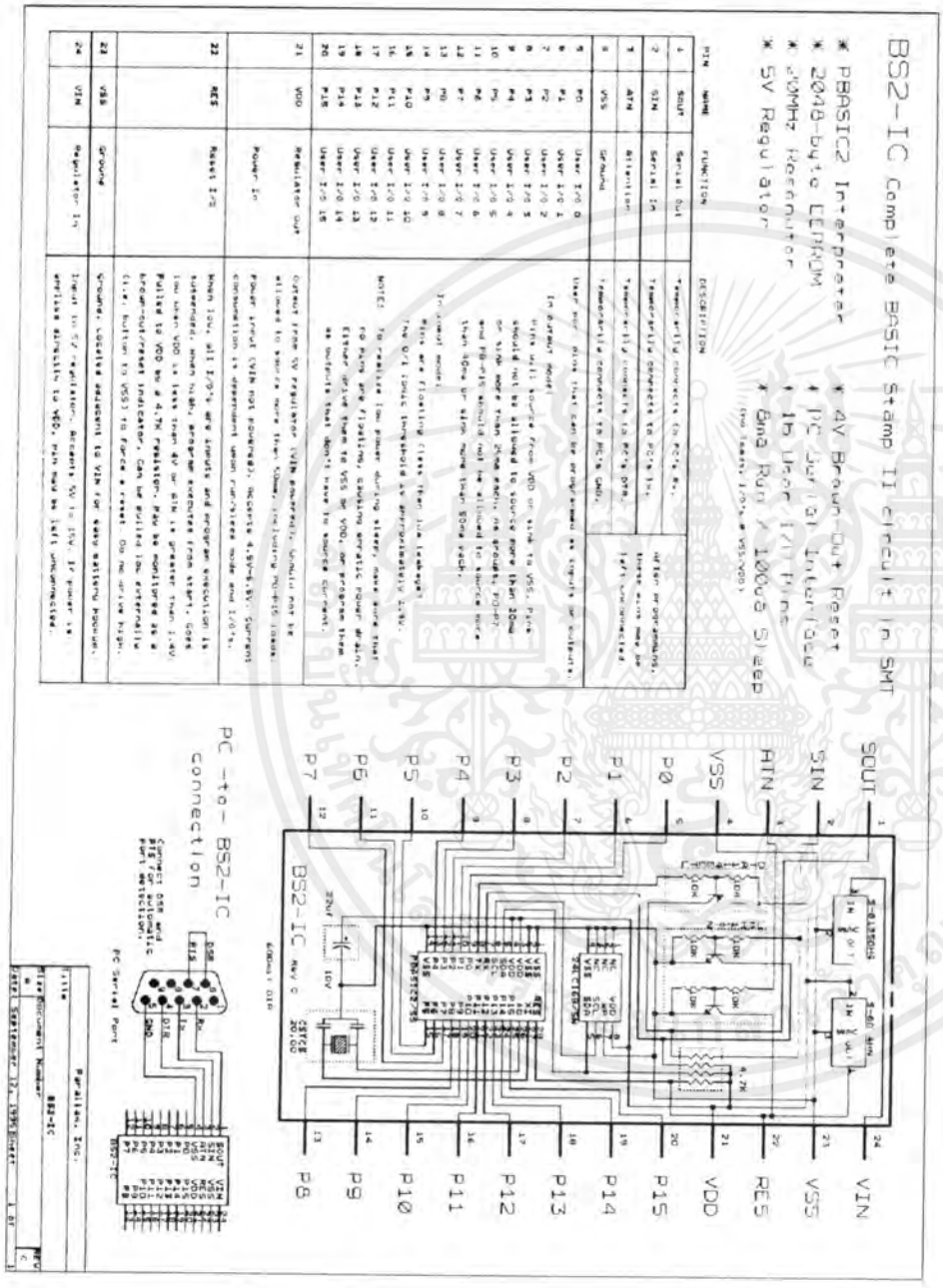
2. If PINS or PIN0..PIN7 appears to the left, or to the left and right, of the equal sign, '=', replace PINS with OUTS and PIN0..PIN7 with OUT0..OUT7.

3. If PINS or PIN0..PIN7 appears to the right of the equal sign, '=', replace PINS with INS and PIN0..PIN7 with IN0..IN7.

4. If PORT appears in an assignment, determine which byte (PINS or DIRS) is affected and replace PORT with the corresponding Stamp II symbol (INS, OUTS or DIRS). If both bytes are affected, separate assignment statements may be needed to accomplish the equivalent effect in the Stamp II.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Schematics



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2003 THRU 2024
HIGH-VOLTAGE, HIGH-CURRENT DARLINGTON ARRAYS

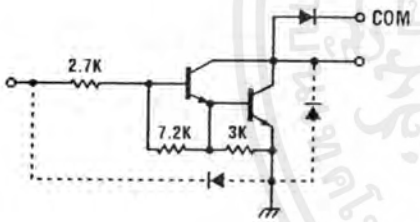
DEVICE PART NUMBER DESIGNATION

$V_{CE(MAX)}$	50 V	95 V
$I_{C(MAX)}$	500 mA	500 mA
Logic	**Part Number**	
5V TTL, CMOS	ULN2003A* ULN2003L*	ULN2023A* ULN2023L
6-15 V CMOS, PMOS	ULN2004A* ULN2004L*	ULN2024A ULN2024L

* Also available for operation between -40°C and +85°C. To order, change prefix from "ULN" to "ULQ".

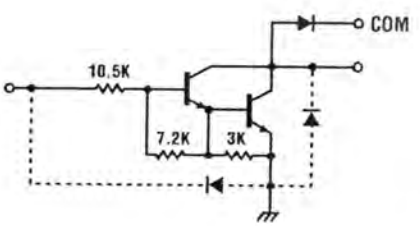
PARTIAL SCHEMATICS

ULN20x3A/L (Each Driver)

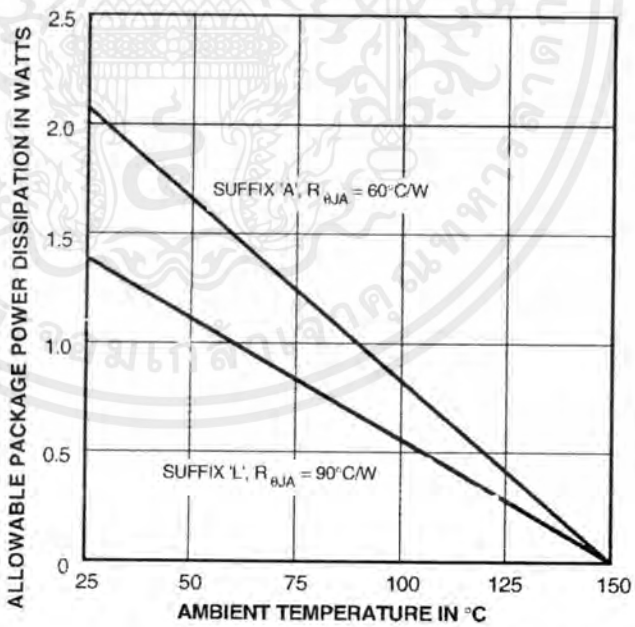


Dwg. No. A-9851

ULN20x4A/L (Each Driver)



Dwg. No. A-9898A



Dwg. GP-006A

X = Digit to identify specific device. Specification shown applies to family of devices with remaining digits as shown. See matrix above.

Allegro MicroSystems, Inc.
115 Northeast Cutoff, Box 15036
Worcester, Massachusetts 01615-0036 (508) 853-5000

Copyright © 1974, 1998 Allegro MicroSystems, Inc.

ไม่วากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2003 THRU 2024
HIGH-VOLTAGE, HIGH-CURRENT DARLINGTON ARRAYS

Types ULN2023A, ULN2023L, ULN2024A, and ULN2024L
ELECTRICAL CHARACTERISTICS at +25°C (unless otherwise noted).

Characteristic	Symbol	Test Fig.	Applicable Devices	Test Conditions	Limits			
					Min.	Typ.	Max.	Units
Output Leakage Current	I_{CEX}	1A	All	V_{CE} = 95 V, T_A = 25°C	—	< 1	50	µA
				V_{CE} = 95 V, T_A = 70°C	—	< 1	100	µA
		1B	ULN2024A/L	V_{CE} = 95 V, T_A = 70°C, V_{IN} = 1.0 V	—	< 5	500	µA
Collector-Emitter Saturation Voltage	$V_{CE(SAT)}$	2	All	I_C = 100 mA, I_B = 250 µA	—	0.9	1.1	V
				I_C = 200 mA, I_B = 350 µA	—	1.1	1.3	V
				I_C = 350 mA, I_B = 500 µA	—	1.3	1.6	V
Input Current	$I_{IN(ON)}$	3	ULN2023A/L	V_{IN} = 3.85 V	—	0.93	1.35	mA
			ULN2024A/L	V_{IN} = 5.0 V	—	0.35	0.5	mA
				V_{IN} = 12 V	—	1.0	1.45	mA
	$I_{IN(OFF)}$	4	All	I_C = 500 µA, T_A = 70°C	50	65	—	µA
Input Voltage	$V_{IN(ON)}$	5	ULN2023A/L	V_{CE} = 2.0 V, I_C = 200 mA	—	—	2.4	V
				V_{CE} = 2.0 V, I_C = 250 mA	—	—	2.7	V
				V_{CE} = 2.0 V, I_C = 300 mA	—	—	3.0	V
			ULN2024A/L	V_{CE} = 2.0 V, I_C = 125 mA	—	—	5.0	V
				V_{CE} = 2.0 V, I_C = 200 mA	—	—	6.0	V
				V_{CE} = 2.0 V, I_C = 275 mA	—	—	7.0	V
				V_{CE} = 2.0 V, I_C = 350 mA	—	—	8.0	V
Input Capacitance	C_{IN}	—	All		—	15	25	pF
Turn-On Delay	t_{PLH}	8	All	0.5 E_{IN} to 0.5 E_{OUT}	—	0.25	1.0	µs
Turn-Off Delay	t_{PHL}	8	All	0.5 E_{IN} to 0.5 E_{OUT}	—	0.25	1.0	µs
Clamp Diode Leakage Current	I_R	6	All	V_R = 95 V, T_A = 25°C	—	—	50	µA
				V_R = 95 V, T_A = 70°C	—	—	100	µA
Clamp Diode Forward Voltage	V_F	7	All	I_F = 350 mA	—	1.7	2.0	V

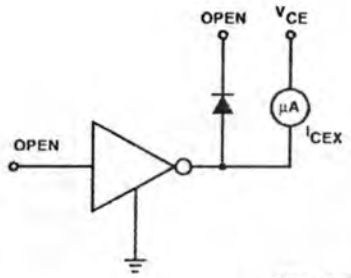
Complete part number includes suffix to identify package style: A = DIP, L = SOIC.

Allegro MicroSystems, Inc.
115 Northeast Cutoff, Box 15036
Worcester, Massachusetts 01615-0036 (508) 853-5000

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

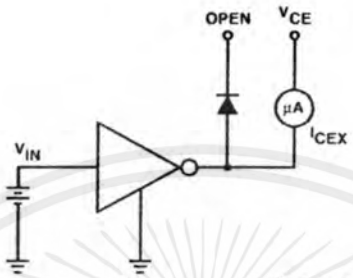
TEST FIGURES

FIGURE 1A



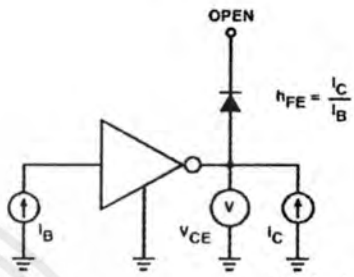
Dwg. No. A-9729A

FIGURE 1B



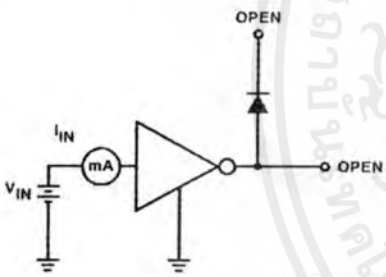
Dwg. No. A-9730A

FIGURE 2



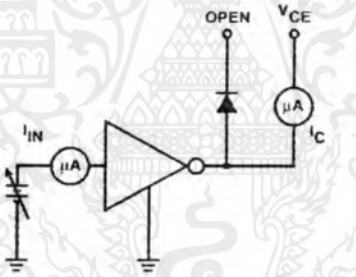
Dwg. No. A-9731A

FIGURE 3



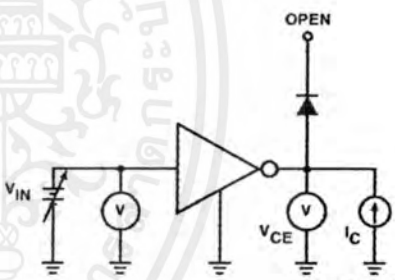
Dwg. No. A-9732A

FIGURE 4



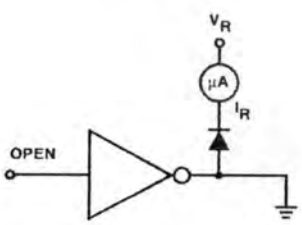
Dwg. No. A-9733A

FIGURE 5



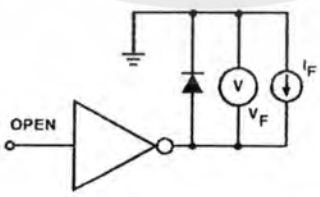
Dwg. No. A-9734A

FIGURE 6



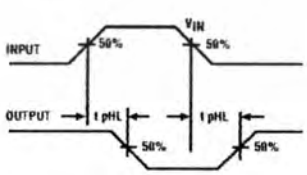
Dwg. No. A-9735A

FIGURE 7

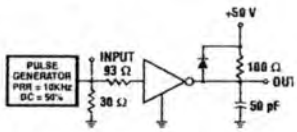


Dwg. No. A-9736A

FIGURE 8



	V_{in}
ULN20X3*	3.5 V
ULN20X4*	12 V



* Complete part number includes a final letter to indicate package.

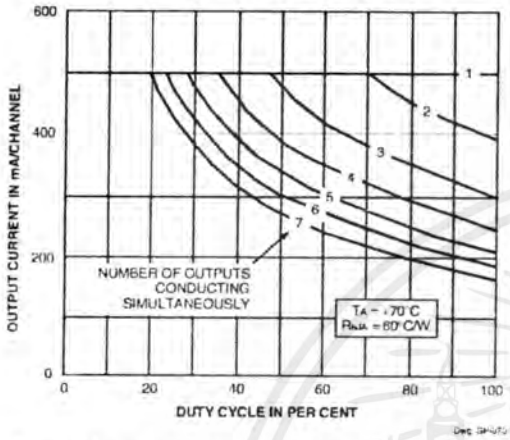
X = Digit to identify specific device. Specification shown applies to family of devices with remaining digits as shown.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

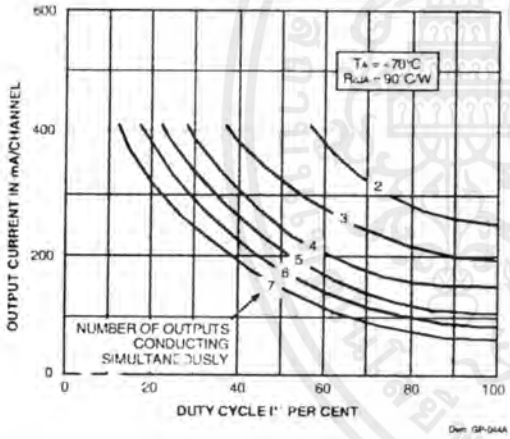
2003 THRU 2024
HIGH-VOLTAGE, HIGH-CURRENT DARLINGTON ARRAYS

ALLOWABLE COLLECTOR CURRENT AS A FUNCTION OF DUTY CYCLE

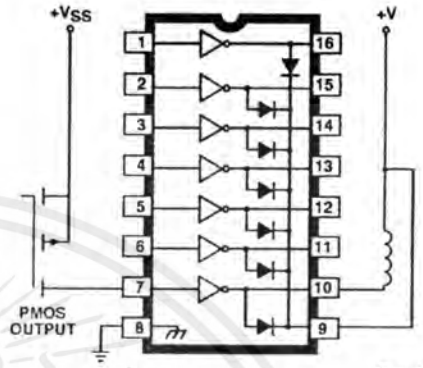
(Dual In-line-Packaged Devices, Suffix 'A')



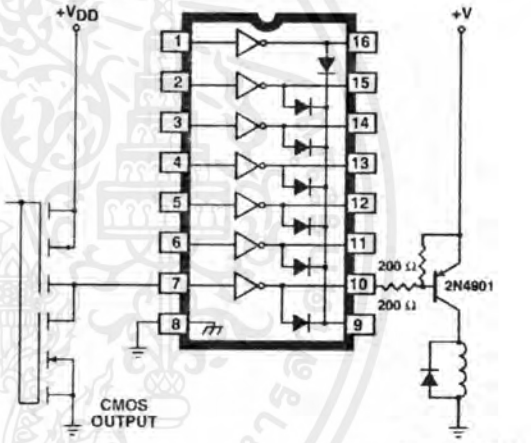
(Small-Outline-Packaged Devices, Suffix 'L')



TYPICAL APPLICATIONS

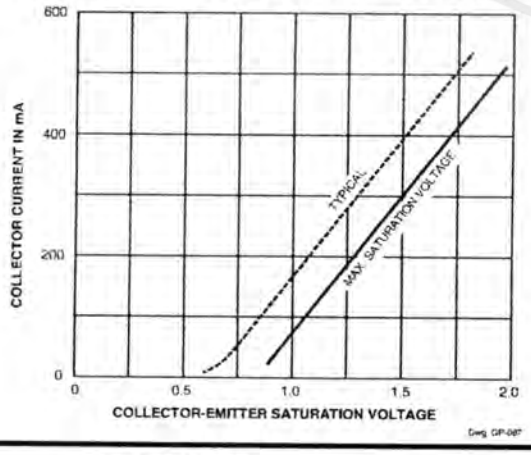


Dwg. No. A-9652

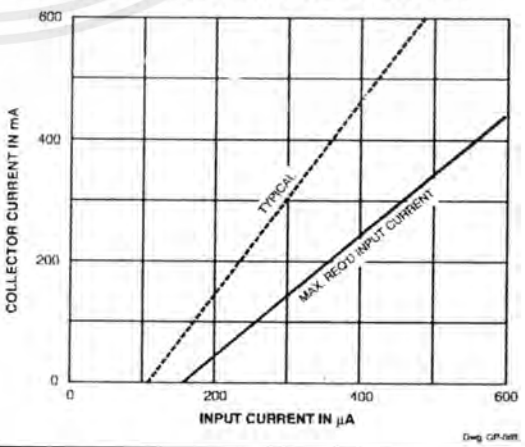


Dwg. No. A-9654A

SATURATION VOLTAGE AS A FUNCTION OF COLLECTOR CURRENT



COLLECTOR CURRENT AS A FUNCTION OF INPUT CURRENT



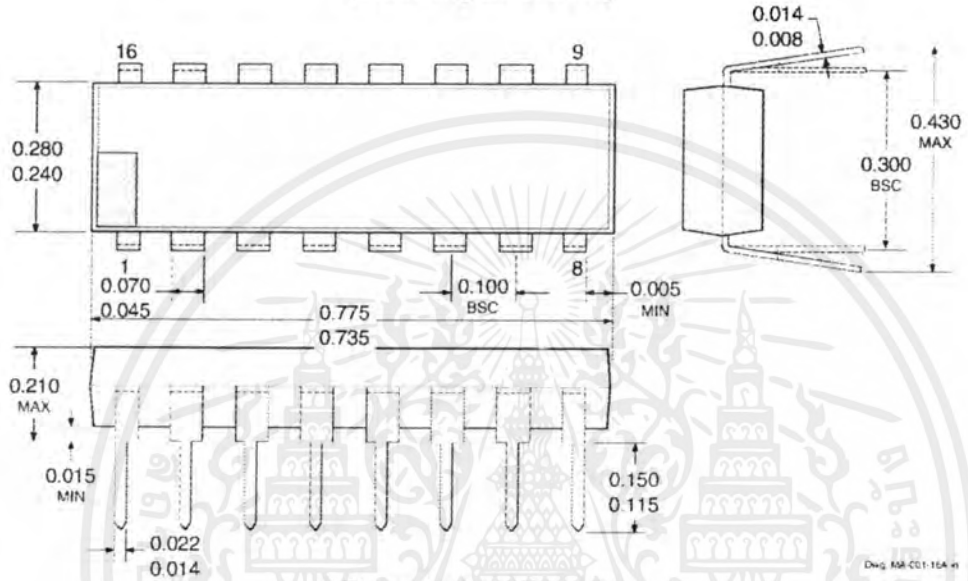
Allegro MicroSystems, Inc.
115 Northeast Cutoff, Box 15036
Worcester, Massachusetts 01615-0036 (508) 853-5000

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

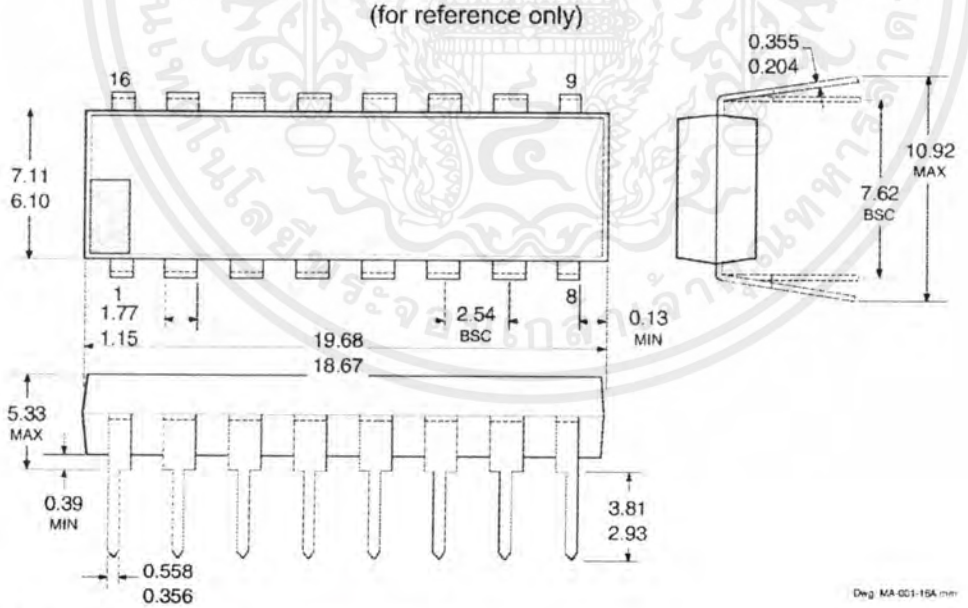
2003 THRU 2024
HIGH-VOLTAGE, HIGH-CURRENT DARLINGTON ARRAYS

PACKAGE DESIGNATOR "A"

Dimensions in Inches
(controlling dimensions)



Dimension in Millimeters
(for reference only)



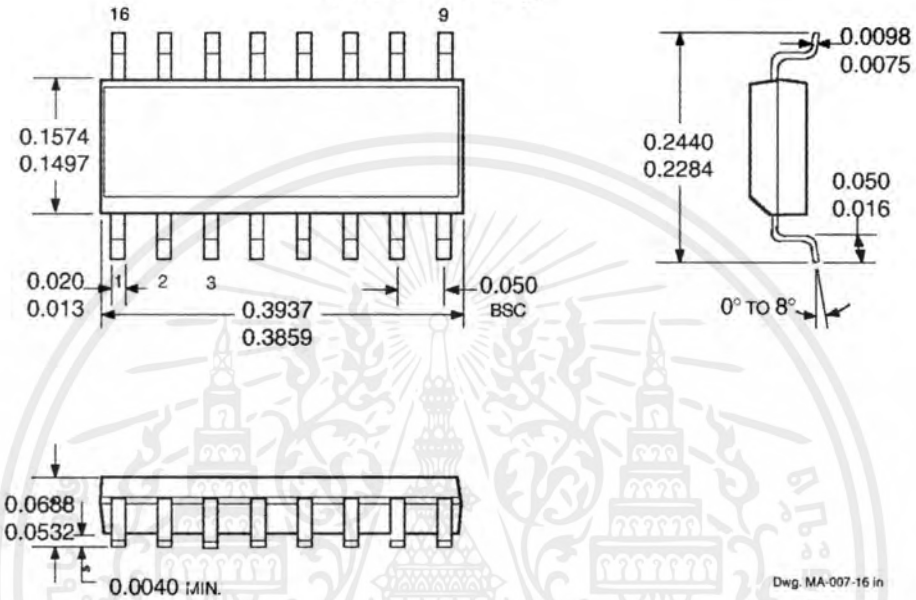
NOTES: 1. Leads 1, 8, 9, and 16 may be half leads at vendor's option.
2. Lead thickness is measured at seating plane or below.
3. Lead spacing tolerance is non-cumulative.
4. Exact body and lead configuration at vendor's option within limits shown.

Allegro MicroSystems, Inc.
115 Northeast Cutoff, Box 15036
Worcester, Massachusetts 01615-0036 (508) 853-5000

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

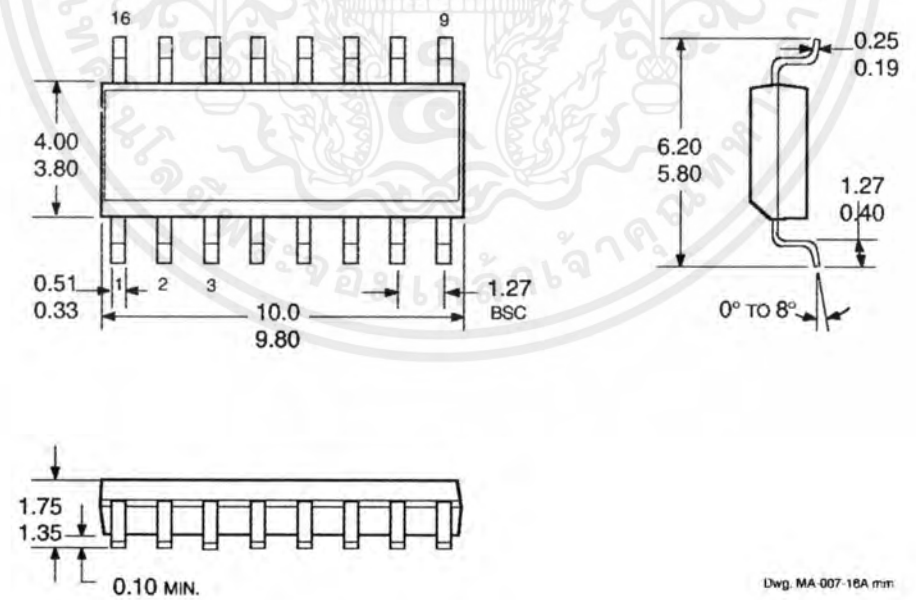
PACKAGE DESIGNATOR "L"

Dimensions in Inches
(for reference only)



Dwg. MA-007-16 in

Dimension in Millimeters
(controlling dimensions)



Dwg. MA-007-16A mm

NOTES: 1. Lead spacing tolerance is non-cumulative.
2. Exact body and lead configuration at vendor's option within limits shown.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้