

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

เครื่องสแกนระบบ 3 มิติ

3D SCAN SYSTEM MACHINE



โดย

นาย คมสัน ถิ่นสุข
นาย ชัดชัย ประสิทธิ์สาท
นาย ฉัฐพงษ์ จันจอม

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2549

๕/๖.
๑/๕ ๒๑
๑๖๔๙

เลขหมู่.....
เลขทะเบียน **72213**
วัน,เดือน,ปี **12 มี.ย. 2550**

b. 117๖51๙๗
i.....

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เครื่องสแกนระบบ 3 มิติ
3D SCAN SYSTEM MACHINE



โดย

นาย กมตัน	ถิ่นสุข	รหัส 47015866
นาย ชัดชัย	ประสิทธิ์สาท	รหัส 47015870
นาย ฉัฐพงษ์	จันทอม	รหัส 47015871

อาจารย์ที่ปรึกษา
ดร. สิริเดช บุญแสง

ปริญญาบัตรสำหรับปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาอิเล็กทรอนิกส์
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2549

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ ปีการศึกษา 2549

ภาควิชา วิศวกรรมอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง เครื่องสแกนระบบ 3 มิติ

ผู้จัดทำ

1. นาย คมตัน ถิ่นสุข รหัส 47015866
2. นาย ชัดชัย ประสิทธิ์สาท รหัส 47015870
3. นาย ณัฐพงษ์ จันทอม รหัส 47015871



อาจารย์ที่ปรึกษา
(อาจารย์ ดร. สิริเดช บุญแสง)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เครื่องสแกนระบบ 3 มิติ

นาย คมสัน ถิ่นสุข รหัส 47015866
นาย ชัคชัย ประสิทธิ์สาธ รหัส 47015870
นาย ฉัฐพงษ์ จันจอม รหัส 47015871
ดร. ศิริเดช บุญแสง อาจารย์ที่ปรึกษา
ปีการศึกษา 2549

บทคัดย่อ

รายงานฉบับนี้อธิบายถึงการออกแบบและการสร้างเครื่องสแกนระบบ 3 มิติ โดยเครื่องสแกนระบบ 3 มิติ ดังกล่าวประกอบด้วยส่วนที่สำคัญ 2 ส่วนคือ ส่วนของซอฟต์แวร์ และส่วนของฮาร์ดแวร์ ในส่วนของซอฟต์แวร์ จะเป็นการเขียนโปรแกรมควบคุมการทำงานและสั่งงานของเครื่องสแกนระบบ 3 มิติ ลงบนโปรแกรม LabVIEW 7.0 ติดต่อกับไมโครคอนโทรลเลอร์ผ่านทางพอร์ตอนุกรม โดยที่จะสามารถควบคุมความเร็วของ สเต็ปมอเตอร์ ทั้ง 3 แกน ได้ ตามต้องการ ส่วนของฮาร์ดแวร์ จะเป็นส่วนของระบบแมคคานิค ประกอบด้วยแกน 3 แกนคือ แกนX, แกนY, แกนZ เคลื่อนที่อิสระจากกัน ควบคุมด้วยสเต็ปมอเตอร์โดยมีชุดไดร์ฟเวอร์ขับจากไมโครคอนโทรลเลอร์ ใช้การส่งกำลังด้วยสกรู โดยมีสไลด์บาร์เป็นไกด์เส้นทางเคลื่อนที่

3D SCAN SYSTEM MACHINE

Mr. Komsun Thinsuk ID. 47015866

Mr. Chudchai Prasitsart ID. 47015870

Mr. Natthapong Chanjorm ID. 47015871

Dr. Siridech Boonsang Advisor

Educational year 2006

Abstract

This report describes a design and construction of 3D Scan System Machine. This machine composes of two important main parts which are software and hardware. The software will control and command of 3D scan system machine. The software is written in LabVIEW 7.0 program. LabVIEW program will contact to microcontroller through serial port of computer. It also control velocity of step motor on 3 axis according to user adjust on LabVIEW program. The hardware is mechatronic system that consist of 3 axis which are X-axis Y-axis and Z-axis, independent from each other axis. They are controlled by step motor with a driver circuit from microcontroller. The screw with a slide bar is to guide the system.

กิตติกรรมประกาศ

ขอขอบพระคุณท่านอาจารย์ ดร.ศิริเดช บุญแสง ผู้เป็นที่ปรึกษาของโครงการนี้ สำหรับคำแนะนำต่างๆ รวมทั้งแนวคิดและค่าใช้จ่ายในการสร้างโครงการ

ขอขอบพระคุณอาจารย์ภาควิชาอิเล็กทรอนิกส์ทุกท่านที่ได้ให้ความรู้ต่างๆ กับพวกกระผม

ขอขอบพระคุณ คุณพ่อ และคุณแม่ ที่ได้อบรมเลี้ยงดูมาเป็นอย่างดี

ขอขอบคุณเพื่อน ๆ ทุกคนที่ให้ความสนใจ และให้การสนับสนุนในการทำโครงการนี้

นาย คมสัน ถิ่นสุข

นาย ชัชชัย ประสิทธิ์สาท

นาย ฉัฐพงษ์ จันจอม



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทคัดย่อ	I
Abstract	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญภาพ	VI
สารบัญตาราง	VII
บทที่ 1 บทนำ	1
บทที่ 2 ทฤษฎีโปรแกรม LabVIEW	3
2.1 บทนำ	3
2.2 ส่วนประกอบของ LabVIEW	4
2.2.1 Front Panel	4
2.2.2 Block Diagram	6
2.2.3 Icon and Connector Pane	7
2.3 เครื่องมือต่างๆ ที่ใช้ในการออกแบบ	8
2.3.1 Tools Palette	8
2.3.2 Controls Palette	9
2.3.3 Functions Palctte	11
2.4 ประเภทของข้อมูล (Data Type)	13
2.4.1 ประเภทของตัวแปร	13
2.4.2 ประเภทของสายสัญญาณ	15
บทที่ 3 ทฤษฎีไมโครคอนโทรลเลอร์ PIC16F87x	16
3.1 โครงสร้างทางฮาร์ดแวร์ของไมโครคอนโทรลเลอร์ PIC16F87X	16
3.2 โครงสร้างของไมโครคอนโทรลเลอร์ PIC16F87x	16
3.3 คุณสมบัติทางเทคนิคของ PIC16F87x/87xA	19
3.4 โหมดสัญญาณนาฬิกา	21
3.5 กระบวนการรีเซตใน PIC16F87x	22
3.6 การรีเซตที่ขา \overline{MCLR}	24

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

	หน้า
3.7 พอร์ตอินพุตเอาต์พุตของไมโครคอนโทรลเลอร์ PIC16F87x	25
3.7.1 พอร์ต A	25
3.7.2 พอร์ต B	29
3.7.3 พอร์ต C	32
3.7.4 พอร์ต D	33
3.7.5 พอร์ต E	34
บทที่ 4 ทฤษฎีการทำงานของสเต็ปมอเตอร์	38
4.1 บทนำ	38
4.2 ชนิดของสเต็ปมอเตอร์	39
4.2.1 แบบแม่เหล็กถาวร (Permanent Magnet : PM)	40
4.2.2 แบบเปลี่ยนแปลงค่ารีลักแทนซ์ (Variable Reluctance : VR)	40
4.2.3 แบบผสม (Hybrid : HB)	41
4.3. การพันขดลวดของสเต็ปมอเตอร์	42
4.3.1 แบบไบโพลาร์ (Bipolar)	42
4.3.2 แบบยูนิโพลาร์ (Unipolar)	43
4.4. การควบคุมการหมุนของสเต็ปมอเตอร์	44
4.4.1 แบบฟลูสเต็ม 1 เฟส (Full – Step 1 Phase)	44
4.4.2 แบบฟลูสเต็ม 2 เฟส (Full – Step 2 Phase)	45
4.4.3 แบบฮาล์ฟสเต็ป (Half Step)	45
4.5 การควบคุมการทำงาน of สเต็ปมอเตอร์	46
บทที่ 5 การออกแบบและการทำงานของวงจร	47
5.1 รายละเอียดของไอซี L297	48
5.2 รายละเอียดของไอซี L298	53

สารบัญ (ต่อ)

	หน้า
5.3 วงจรควบคุมและขับเคลื่อนมอเตอร์	59
5.3.1 การทำงานของวงจรควบคุมและวงจรขับเคลื่อนมอเตอร์	60
5.4 วงจรไมโครคอนโทรลเลอร์	61
5.4.1 การทำงานในส่วนของไมโครคอนโทรลเลอร์	62
5.5 การออกแบบโปรแกรมควบคุมไมโครคอนโทรลเลอร์	63
5.5.1 การทำงานในส่วนของโปรแกรม	70
บทที่ 6 กลไกการขับเคลื่อนของเครื่องสแกน 3 มิติ	83
6.1 หลักการโดยทั่วไปในการออกแบบส่วนประกอบต่างๆ	83
6.2 ส่วนประกอบสำคัญในการสร้างเครื่องสแกน 3 มิติ	84
6.3 การทำงานของเครื่องสแกน 3 มิติ	84
6.4 แสดงรูปโครงสร้างส่วนต่าง ๆ ของระบบ (Mechanic)	84
บทที่ 7 ผลการทดลอง	74
7.1 การวิเคราะห์สัญญาณจากวงจรไมโครคอนโทรลเลอร์และวงจรไครฟมอเตอร์	74
7.2 การหาระยะในการเคลื่อนที่ต่อการขับเคลื่อน 1 STEP	78
7.3 การทดลองในส่วนของโปรแกรม LabVIEW	80
7.4 Source Code สำหรับไมโครคอนโทรลเลอร์	88
ปัญหาที่พบในการดำเนินงาน	114
สรุปผลการทดลอง	114
บทที่ 8 บทสรุป	115
บรรณานุกรม	
ภาคผนวก	

สารบัญภาพ

	หน้า
รูปที่ 2.1 แสดงส่วนของหน้าต่าง Front Panel	5
รูปที่ 2.2 แสดงส่วนของหน้าต่าง Block Diagram	7
รูปที่ 2.3 แสดง Tools Palette	8
รูปที่ 2.4 แสดง Controls Palette	10
รูปที่ 2.5 แสดง Functions Palette	12
รูปที่ 2.6 แสดงตัวอย่างของตัวแปรต่างๆ ที่ใช้ใน LabVIEW	14
รูปที่ 2.7 แสดงประเภทของสายสัญญาณแบบต่างๆ ใน LabVIEW	15
รูปที่ 3.1 ไดอะแกรมแสดงรูปแบบสถาปัตยกรรมของไมโครคอนโทรลเลอร์แบบฮาร์ดแวร์	16
รูปที่ 3.2 (ก) โครงสร้างการทำงานของไมโครคอนโทรลเลอร์ PIC16F87x	17
รูปที่ 3.2 (ข) โครงสร้างการทำงานของไมโครคอนโทรลเลอร์ PIC16F87x รุ่น 40 ขา	18
รูปที่ 3.3 การจัดขาของไมโครคอนโทรลเลอร์ PIC16F87x	19
รูปที่ 3.4 การต่อตัวต้านทานและตัวเก็บประจุเพื่อกำหนดความถี่สัญญาณนาฬิกาเมื่อไมโครคอนโทรลเลอร์ PIC16F87X ทำงานในโหมด RC	21
รูปที่ 3.5 แสดงกลไกของการเกิดรีเซตในไมโครคอนโทรลเลอร์ PIC16F87X	23
รูปที่ 3.6 วงจรเพาเวอร์อนรีเซตสำหรับไมโครคอนโทรลเลอร์ PIC16F87x	24
รูปที่ 3.7 โครงสร้างขา RA0-RA3 ของพอร์ต A ในไมโครคอนโทรลเลอร์ PIC16F877x	26
รูปที่ 3.8 โครงสร้างขา RA4 ของพอร์ต A	28
รูปที่ 3.9 โครงสร้างขาพอร์ต RA5 ของพอร์ต A ในไมโครคอนโทรลเลอร์ PIC16F87x	29
รูปที่ 3.10 โครงสร้างของขาพอร์ต B ในไมโครคอนโทรลเลอร์ PIC16F87x	32
รูปที่ 3.11 โครงสร้างของขาพอร์ต D ในไมโครคอนโทรลเลอร์ PIC16F87x	34
รูปที่ 3.12 โครงสร้างของขาพอร์ต E	35

สารบัญภาพ (ต่อ)

	หน้า
รูปที่ 4.1 โครงสร้างของสเต็ปิ่งมอเตอร์แบบแม่เหล็กถาวร	40
รูปที่ 4.2 โครงสร้างของสเต็ปิ่งมอเตอร์แบบเปลี่ยนแปลงค่ารีลักแทนซ์และสัญลักษณ์	41
รูปที่ 4.3 แสดงสเต็ปิ่งมอเตอร์แบบผสมขนาด 5 เฟส	41
รูปที่ 4.4 แสดงการพันขดลวดและวงจรสวิทช์แบบ Bipolar	42
รูปที่ 4.5 แสดงการพันขดลวดและวงจรสวิทช์แบบ Unipolar	43
รูปที่ 4.6 แสดงสัญลักษณ์การต่อสเต็ปิ่งมอเตอร์แบบต่างๆ	44
รูปที่ 4.7 แสดงไดอะแกรมการควบคุมสเต็ปิ่งมอเตอร์	46
รูปที่ 5.1 โครงสร้างการทำงานในส่วนต่างๆ	47
รูปที่ 5.2 ตำแหน่งขาของไอซี L297	48
รูปที่ 5.3 โครงสร้างภายในไอซี L297	49
รูปที่ 5.4 สัญญาที่ใช้จับในโหมดฮาลฟสเต็ป	49
รูปที่ 5.5 สัญญาที่ใช้จับในโหมด Two Phase	50
รูปที่ 5.6 สัญญาที่ใช้จับในโหมด One Phase	50
รูปที่ 5.7 แสดงรายละเอียดของขาต่างๆของไอซี L297	51
รูปที่ 5.8 คุณสมบัติทางไฟฟ้าของไอซี L297	52
รูปที่ 5.9 แสดงความสัมพันธ์ของสัญญา	53
รูปที่ 5.10 โครงสร้างภายในไอซี L298	54
รูปที่ 5.11 ค่าสูงสุดที่ใช้ในการทำงานของ L298	54
รูปที่ 5.12 ตำแหน่งขาของไอซี L298	55
รูปที่ 5.13 คุณสมบัติของขาต่างๆของไอซี L298	55
รูปที่ 5.14 คุณสมบัติทางไฟฟ้าของไอซี L298	56
รูปที่ 5.15 แสดงค่าแรงดันอิมิตัวกับกระแสเอาต์พุต	57
รูปที่ 5.16 การทดสอบการทำงานเบื้องต้นของไอซี L298	57
รูปที่ 5.17 ค่ากระแส Delay ต่อสัญญา Input	58
รูปที่ 5.18 การต่อใช้งานระหว่างไอซี L297 กับไอซี L298	58
รูปที่ 5.19 วงจรควบคุมและวงจรจับสเต็ปมอเตอร์	59

สารบัญภาพ (ต่อ)

	หน้า
รูปที่ 5.20 วงจรไมโครคอนโทรลเลอร์ที่ใช้ติดต่อกับคอมพิวเตอร์	61
รูปที่ 5.21 ส่วนของออปโตที่ใช้ในการเชื่อมโยงระหว่างไมโครคอนโทรลเลอร์กับบอร์ดควบคุมและขับเคลื่อนมอเตอร์	61
รูปที่ 5.22 ส่วนของวงจรที่ใช้ในการติดต่อบetweenไมโครคอนโทรลเลอร์กับคอมพิวเตอร์	62
รูปที่ 5.23 โฟลวชาร์ตโปรแกรมหลัก	63
รูปที่ 5.24 โฟลวชาร์ตโปรแกรมย่อยบริการอินเตอร์รัพจากพอร์ตอนุกรม	64
รูปที่ 5.24 ออกจากโปรแกรมย่อยบริการอินเตอร์รัพจากพอร์ต RB0 (ต่อ)	64-1
รูปที่ 5.25 โฟลวชาร์ตโปรแกรมย่อยการหาจุดเริ่มต้น	65
รูปที่ 5.26 โฟลวชาร์ตโปรแกรมย่อยการเคลื่อนที่แกน X,Y,Z ไปข้างหน้า	66
รูปที่ 5.27 โฟลวชาร์ตโปรแกรมย่อยการเคลื่อนที่แกน X,Y,Z ไปข้างหน้า	67
รูปที่ 5.28 โฟลวชาร์ตโปรแกรมย่อยการเคลื่อนที่แกน X,Y,Z ไปข้างหน้า	68
รูปที่ 5.29 โฟลวชาร์ตโปรแกรมย่อยการสแกน	69
รูปที่ 6.1 สกรูยาวใช้สำหรับขับเคลื่อน	72
รูปที่ 6.2 ชุดขับเคลื่อนแกน Z	73
รูปที่ 6.3 แสดงชุดขับเคลื่อนทั้ง 3 แกน ของเครื่องสแกนระบบ 3 มิติ	73
รูปที่ 7.1 CH1 สัญญาณที่ออกมาจากไมโครคอนโทรลเลอร์ CH2 สัญญาณที่ออกมาจากออปโต ที่คาบเวลาเท่ากับ 1.8 ms.	74
รูปที่ 7.2 CH1 สัญญาณที่ออกมาจากไมโครคอนโทรลเลอร์ CH2 สัญญาณที่ออกมาจากออปโตที่คาบเวลาเท่ากับ 2 ms.	75
รูปที่ 7.3 CH1 สัญญาณเฟส A , CH2 สัญญาณเฟส B ที่ออกมาจากไอซี L297 เมื่อป้อนคล็อกคาบเวลาเท่ากับ 2 ms.	75
รูปที่ 7.4 CH1 สัญญาณเฟส C และ CH2 สัญญาณเฟส D ที่ออกมาจากไอซี L297 เมื่อป้อนคล็อกคาบเวลาเท่ากับ 2 ms.	76

สารบัญภาพ (ต่อ)

หน้า

รูปที่ 7.5 สัญญาณที่ได้จากไอซี L297 เมื่อวัดระหว่างเฟส A กับ B ร่วมกัน เมื่อป้อนคล็อกคาบเวลาเท่ากับ 2 ms	76
รูปที่ 7.6 สัญญาณที่ได้จากไอซี L297 เมื่อวัดระหว่างเฟส C กับ D ร่วมกัน เมื่อป้อนคล็อกคาบเวลาเท่ากับ 2 ms.	77
รูปที่ 7.7 แสดงการใช้เวอร์เนียร์วัดระยะการเคลื่อนที่	79
รูปที่ 7.8 แสดงส่วนของ Front Panel ของโปรแกรม LabVIEW ที่ใช้งาน	80
รูปที่ 7.9 แสดงส่วนของ Front Panel และ Block Diagram	81
รูปที่ 7.10 แสดงส่วนของ Front Panel และ Block Diagram ในส่วนของการหยุดการทำงาน (Stop)	82
รูปที่ 7.11 แสดงส่วนของ Front Panel และ Block Diagram ในส่วนของการเคลื่อนที่ไปข้างหน้า (Forward)	83
รูปที่ 7.12 แสดงส่วนของ Front Panel และ Block Diagram ในส่วนของการเคลื่อนที่ย้อนกลับ (Backward)	84
รูปที่ 7.13 แสดงส่วนของ Front Panel และ Block Diagram ในส่วนของการสแกน (Scan)	86

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

	หน้า
ตารางที่ 2.1 แสดงรายละเอียดของตัวแปรต่างๆ (Numeric Control และ Indicators)	14
ตารางที่ 3.1 แสดงความสัมพันธ์ระหว่างบิตแสดงสถานะกับลักษณะการเกิดรีเซตขึ้นของไมโครคอนโทรลเลอร์ PIC16F87x	23
ตารางที่ 3.2 แสดงหน้าที่การทำงานของขาพอร์ต E	37
ตารางที่ 4.1 แสดงการควบคุมสเต็ปिंगมอเตอร์แบบ Full – Step 1 Phase	45
ตารางที่ 4.2 แสดงการควบคุมสเต็ปिंगมอเตอร์แบบ Full – Step 2 Phase	45
ตารางที่ 4.3 แสดงการควบคุมสเต็ปึงมอเตอร์แบบ Half Phase	46
ตารางที่ 7.1 การหาระยะการเคลื่อนที่ต่อการขับเคลื่อน 1 STEP	78

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

โครงการได้นำเสนอการออกแบบและควบคุมการทำงานของเครื่องสแกนระบบ 3 มิติ ด้วยวงจรอิเล็กทรอนิกส์และคอมพิวเตอร์ โดยเครื่องสแกน 3 มิติ ประกอบด้วย สเต็ปปีงมอเตอร์แบบไบโพลาาร์ จำนวน 3 ตัว ที่ทำงานได้อย่างอิสระเพื่อควบคุมแกน X, Y, Z โดยที่เครื่องสแกนระบบ 3 มิติ จะมีขนาดเล็กขนย้ายได้สะดวก มีความคงทนในการใช้งานและสามารถทำงานได้อย่างมีประสิทธิภาพ ส่วนการควบคุมฮาร์ดแวร์ ให้มีทิศทางการเคลื่อนที่และความเร็ว ได้อย่างถูกต้องนั้น ทำโดยใช้การควบคุมด้วยเครื่องคอมพิวเตอร์ ทางคณะผู้จัดทำได้ใช้โปรแกรม LabVIEW ในการเขียนโปรแกรมควบคุมและแสดงผลทางหน้าจอคอมพิวเตอร์ ติดต่อกับชุดไมโครคอนโทรลเลอร์ PIC16F877 ซึ่งใช้การเชื่อมต่อกับคอมพิวเตอร์ผ่านพอร์ตอนุกรม และการควบคุมวงจรสเต็ปเปอร์มอเตอร์ทั้ง 3 แกน ได้ใช้วงจรขับสเต็ปเปอร์มอเตอร์ที่รับการสั่งงานมาจากชุดไมโครคอนโทรลเลอร์ PIC16F877 อีกที

เครื่องสแกนระบบ 3 มิติ ทำงานอัตโนมัติด้วยวงจรอิเล็กทรอนิกส์และคอมพิวเตอร์ ซึ่งเชื่อมต่อกันระหว่างคอมพิวเตอร์กับอุปกรณ์ฮาร์ดแวร์ ซึ่งเราใช้พอร์ตอนุกรมเป็นตัวส่งผ่านสัญญาณควบคุมสเต็ปปีงมอเตอร์ และรับสัญญาณ Limit Switch จากอุปกรณ์ฮาร์ดแวร์ จากโครงการนี้คณะผู้จัดทำได้ทำการศึกษาและทำการทดลองประกอบด้วย 2 ส่วนด้วยกันคือ ฮาร์ดแวร์ และ ซอฟต์แวร์

โดยในส่วนของ ซอฟต์แวร์ นั้นได้ทำการเขียนโปรแกรมด้วยโปรแกรม LabVIEW สั่งงานควบคุมและแสดงผลทางหน้าจอคอมพิวเตอร์ โดยติดต่อกับชุดไมโครคอนโทรลเลอร์ PIC16F877 ผ่านทางพอร์ตอนุกรม โดยที่จะสามารถควบคุมความเร็วของ สเต็ปเปอร์มอเตอร์ ทั้ง 3 แกนได้ ตามต้องการ

ใน ส่วนของฮาร์ดแวร์จะเป็นส่วนของระบบแมคคานิก ประกอบด้วยแกน 3 แกนคือ แกน X, แกน Y, แกน Z เคลื่อนที่อิสระจากกัน ควบคุมด้วยสเต็ปเปอร์มอเตอร์โดยมีชุดไคร์ฟเวอร์ขับจากไมโครคอนโทรลเลอร์ ใช้การส่งกำลังด้วยสกรู โดยมีสไลด์บาร์เป็นไกด์เส้นทางการเคลื่อนที่

วัตถุประสงค์ของโครงการ

1. เพื่อศึกษาการทำงานและการออกแบบโครงสร้างของเครื่องสแกน 3 มิติ
2. เพื่อทำการออกแบบและสร้างเครื่องสแกน 3 มิติ โดยให้มีราคาในการสร้างที่ถูกลงกว่าที่ขายกันในท้องตลาด
3. เพื่อศึกษาวิธีการเขียนโปรแกรมบนโปรแกรมแล็บวิว(Lab VIEW) ติดต่อกับไมโครคอนโทรลเลอร์ ผ่านทางพอร์ตอนุกรมในการควบคุมการทำงานของเครื่องสแกน 3 มิติ
4. เพื่อศึกษาวิธีการนำไมโครคอนโทรลเลอร์ PIC16F87x มาใช้ควบคุมการหมุนของสเต็ปเปอร์มอเตอร์
5. เพื่อศึกษาโครงสร้างและการทำงานของสเต็ปเปอร์มอเตอร์ชนิดไบโพลาร์
6. เพื่อศึกษาวิธีการนำไอซีที่มีจำหน่ายในท้องตลาดมาใช้งานเป็นวงจรรับสเต็ปเปอร์มอเตอร์ โดยศึกษาคุณสมบัติต่างๆ จากข้อมูลที่บริษัทให้มา(Data Sheet) แล้วนำมาออกแบบเป็นวงจรที่สามารถใช้งานได้

ประโยชน์ที่จะได้รับ

1. ได้รับความรู้ในการออกแบบและการสร้างโครงสร้างของเครื่องสแกน 3 มิติ
2. สามารถออกแบบกลไกการทำงานและควบคุมการทำงานของเครื่องสแกน 3 มิติได้
3. เข้าใจหลักการการทำงานของสเต็ปเปอร์มอเตอร์ และสามารถนำไปประยุกต์ใช้งานได้ตามที่ต้องการ
4. ได้รับความรู้ทั้งทางด้านฮาร์ดแวร์และซอฟต์แวร์ของไมโครคอนโทรลเลอร์ของตระกูล PIC16F87x
5. ได้รับความรู้ในการสร้างเครื่องรับสเต็ปเปอร์มอเตอร์จากไอซีที่มีขายในท้องตลาด โดยการศึกษาจากข้อมูลที่บริษัทให้มา(Data Sheet) แล้วนำมาออกแบบเป็นวงจรที่สามารถใช้งานได้
6. ได้รับความรู้ในการเขียนโปรแกรมบนโปรแกรมแล็บวิว(Lab VIEW) ติดต่อกับไมโครคอนโทรลเลอร์ ผ่านทางพอร์ตอนุกรม เพื่อใช้ในการควบคุมการทำงานของเครื่องสแกน 3 มิติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ทฤษฎีโปรแกรม LabVIEW

2.1 บทนำ

LabVIEW คือ โปรแกรมที่ใช้ในการพัฒนาโปรแกรมประยุกต์ (Application) ชนิดหนึ่ง เช่นเดียวกับ โปรแกรม Visual Basic และ Visual C++ ซึ่งจะเป็นการเขียนโปรแกรมโดยใช้รูปภาพ (Graphical-based Programming) ซึ่งจะแตกต่างจากแนวคิดของการเขียนโปรแกรมแบบใช้ตัวอักษร (Text based Programming) เช่น ภาษา Basic, C หรือ Pascal เป็นต้น โดยที่การเขียนโปรแกรมในแบบของภาษา Basic, C หรือ Pascal จะทำงานจากบนลงล่าง แต่โปรแกรมที่เขียนหรือพัฒนาขึ้นโดยใช้ LabVIEW จะทำงานโดยอาศัยหลักการของการเคลื่อนที่ของข้อมูล (Dataflow) ซึ่งมีหลักการทำงานดังต่อไปนี้ คือ

1. ฟังก์ชันหรือ SubVI จะทำงานเมื่อมีข้อมูล (Input)
2. เมื่อฟังก์ชันหรือ SubVI ทำงานเสร็จจะให้ข้อมูล (Output) ไปยังฟังก์ชันอื่นหรือ SubVI อื่นที่ต้องการข้อมูล
3. ข้อมูลจะถูกส่งผ่านโดยสาย (Wire)

LabVIEW ย่อมาจาก Laboratory Virtual Instrument Workbench โปรแกรมที่เขียนหรือพัฒนาขึ้นโดยใช้ LabVIEW จะถูกเรียกว่า Virtual Instrument หรือเรียกย่อ ๆ ว่า “ VIs ” ซึ่งจะมี ส่วนที่ควบคุม (Control) และแสดงผล (Indicator) คล้ายคลึงกับเครื่องมือวัดต่างๆ ไป เช่น มีปุ่มปรับค่า , ปุ่มเปิดปิด และกราฟแสดงผล เป็นต้น LabVIEW สามารถเขียนเป็นสมการต่างๆ ที่ซับซ้อนได้ เหมือนกับการเขียนโปรแกรมแบบ Text based Programming โดยที่ LabVIEW จะใช้ Formular Node (เครื่องมือที่ใช้ในการเขียนสมการแบบใช้ตัวอักษร) การทำงานของ LabVIEW จะเน้นทางการติดต่อกับเครื่องมือหรือทรานสดิวเซอร์ (Tranducer) ที่ใช้วัดสัญญาณทางกายภาพต่างๆ และนอกจากนั้น LabVIEW ยังสามารถนำเอาประโยชน์ของ Computer Technology และ Network Technology มาประยุกต์ใช้งาน เช่น การเก็บข้อมูลลงในฐานข้อมูล (Database) หรือการส่งผ่านข้อมูลทางอินเทอร์เน็ต ทำให้ในหลายๆ อุตสาหกรรมได้นำเอา LabVIEW มาประยุกต์ใช้กับงานในด้านของการวัด, ควบคุม และออโตเมชัน (Automation)

ในส่วนของ LabVIEW จะมี Front Panel ซึ่งเปรียบเสมือนกับสิ่งที่ผู้ใช้จะเห็นและควบคุมการทำงาน ผู้ใช้สามารถสร้างรูปแบบในส่วนนี้ขึ้นเองได้อย่างรวดเร็วเพราะ LabVIEW จะมีส่วนประกอบต่างๆ ที่ใช้สำหรับออกแบบหน้าจอมากมาย เช่น จอแสดงผลแบบออสซิลโลสโคป (Oscilloscope), ปุ่มหมุน (Dial), และสวิตช์ (Switch) เป็นต้น และในส่วนของพื้นที่ที่ใช้ในการเขียนโปรแกรมจะเรียกว่า Block Diagram ซึ่งจะเปรียบเสมือนกับอุปกรณ์ (Hardware) ภายในเครื่องมือวัด โดยที่ LabVIEW จะควบคุมการทำงานและแสดงผลผ่านทางคอมพิวเตอร์

LabVIEW จะอาศัยหลักการทำงานของเครื่องมือวัดหรือการวัดคุม ทำให้ผู้ใช้สามารถออกแบบรูปแบบของโปรแกรมได้ตามที่ผู้ใช้ต้องการ โดยหลักการดังกล่าวสามารถแบ่งออกเป็น 3 ส่วนใหญ่ๆ คือ

1. Acquisition ซึ่งเป็นส่วนที่รับข้อมูล (Input) จากสิ่งแวดล้อมภายนอกเข้าสู่ระบบ ในที่นี้คือ คอมพิวเตอร์ โดยข้อมูลที่เข้าสู่ระบบนี้อาจจะมาจาก การ์ด DAQ (สำหรับสัญญาณทางไฟฟ้า), การ์ด IMAQ (สำหรับข้อมูลประเภทรูป) หรือ การ์ด GPIB (สำหรับควบคุมเครื่องมือวัด)
2. หลังจากที่ได้รับข้อมูลแล้วอาจจะผ่านฟังก์ชันในการทำ Analysis หรือวิเคราะห์ข้อมูล ซึ่งจะแสดงผลในรูปแบบที่สื่อความหมายในสิ่งที่ผู้ใช้งานสามารถนำไปแสดงแทนสื่อที่วัดได้และใช้งานได้
3. Presentation คือการแสดงผลในรูปแบบที่เป็นประโยชน์ต่อผู้ใช้งาน โดยอาจจะแสดงผลบนหน้าจอคอมพิวเตอร์ เช่น DMM (Digital Multimeter) เป็นการแสดงผลเฉพาะสัญญาณที่วัดได้โดยไม่จำเป็นต้องรู้ความสัมพันธ์กับเวลา หรือ Oscilloscope เป็นการแสดงผลของข้อมูลที่สัมพันธ์กับเวลา หรือ Spectrum Analyzer เป็นการแสดงผลของสัญญาณในรูปของความถี่หรือการพิมพ์ออกมาเป็นรายงานหรือเก็บข้อมูลในฮาร์ดดิสก์

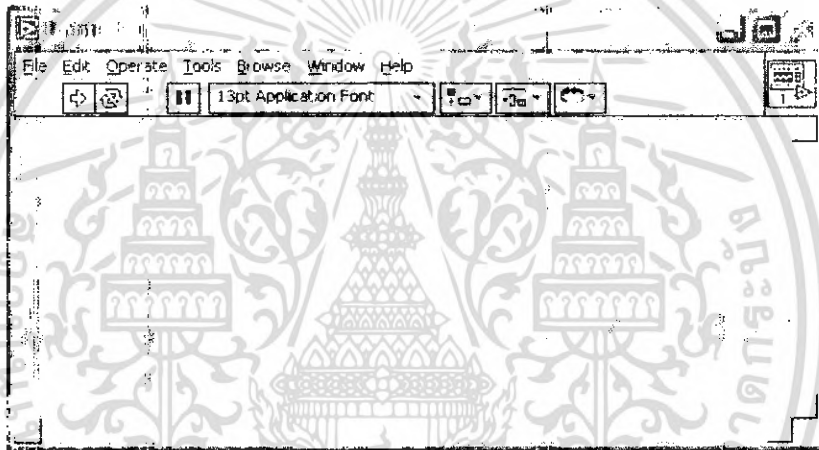
2.2 ส่วนประกอบของ LabVIEW

ดังที่กล่าวไว้ในตอนต้นแล้วว่า โปรแกรมถูกที่เขียนหรือพัฒนาขึ้นโดยใช้ LabVIEW จะถูกเรียกว่า Virtual Instrument หรือเรียกย่อๆ ว่า “VIs” ซึ่งจะประกอบด้วยส่วนประกอบที่สำคัญอยู่ 3 ส่วนดังต่อไปนี้ คือ

- Front Panel
- Block Diagram
- Icon and Connector Pane




2.2.1 Front Panel

Front Panel คือ ส่วนที่ใช้ในการติดต่อกับผู้ใช้งาน (User Interface) เป็นพื้นที่ที่ผู้ใช้งานใช้ในการออกแบบโปรแกรมในการรับค่า, ปรับค่า หรือแสดงผล ในส่วนนี้จะประกอบด้วย Control หรือ Input (รับข้อมูลจากผู้ใช้งาน) และ Indicator หรือ Output (แสดงผลให้ผู้ใช้งานเห็น) ดังนั้นในส่วนของ Front Panel นี้ จะเป็นส่วนที่ผู้ใช้งานใช้ในการควบคุมการทำงานและแสดงผลของตัวโปรแกรม





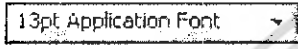







รูปที่ 2.1 แสดงส่วนของหน้าต่าง Front Panel

รายละเอียดของ Front Panel Toolbar

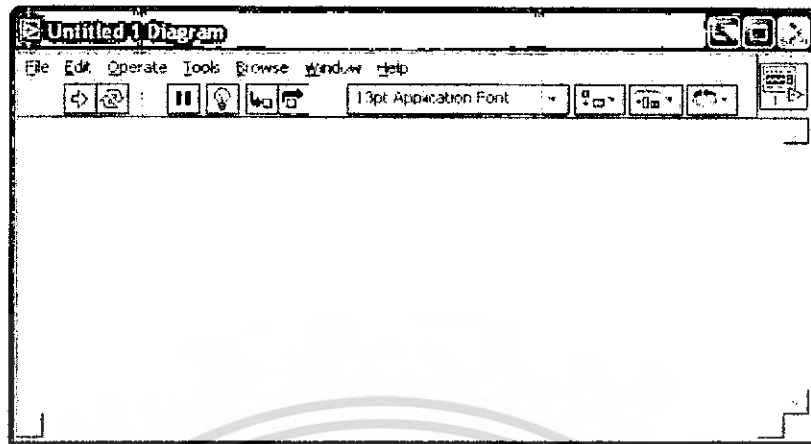
-  Run ใช้สำหรับการเริ่มการทำงานของ โปรแกรมแบบทำงาน 1 รอบ
-  Running เป็นตัวแสดงว่าโปรแกรมกำลังอยู่ในขณะทำงาน
-  List Error เป็นตัวแสดงว่าโปรแกรมมีข้อผิดพลาด (Error)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

-  Run Continuously ใช้สำหรับการเริ่มการทำงานของโปรแกรมแบบทำงานต่อเนื่อง
-  Running Continuously เป็นตัวแสดงว่าโปรแกรมกำลังทำงานแบบต่อเนื่อง
-  Abort Execution ใช้สำหรับการยกเลิกการทำงานของโปรแกรม
-  Pause ใช้สำหรับการพักการทำงานของโปรแกรมชั่วคราว
-  13pt Application Font Font Tool ใช้สำหรับการกำหนดลักษณะของตัวอักษรที่แสดงในโปรแกรม
-  Align Objects ใช้สำหรับการจัดแนวของสิ่งต่างๆ (Objects)
-  Distribute Objects ใช้สำหรับการจัดระยะทางระหว่างสิ่งต่างๆ
-  Resize Objects ใช้สำหรับการปรับขนาดของสิ่งต่างๆ ที่อยู่บน Front Panel หลายๆ สิ่งซ้อนกัน
-  Reorder ใช้สำหรับการจัดลำดับของ Objects ที่วางซ้อนกัน
-  ใช้สำหรับการเรียกใช้ Context Help

2.2.2 Block Diagram

Block Diagram คือ พื้นที่ที่ใช้ในการเขียนหรือพัฒนาโปรแกรม หรือพื้นที่ในส่วนของ Source Code ซึ่งจะเป็นการเขียนโปรแกรมโดยใช้รูปภาพ ในส่วนนี้จะเปรียบเสมือนกับโครงสร้างภายในของเครื่องมือวัด



รูปที่ 2.2 แสดงส่วนของหน้าต่าง Block Diagram

รายละเอียดของ Block Diagram Toolbar

-  Highlight Execution ใช้สำหรับการแสดงการเคลื่อนที่ของข้อมูล
-  Step Info ใช้สำหรับการ Run Program แบบทีละคำสั่ง และลงไปถึงคำสั่งของ SubVI
-  Step Over ใช้สำหรับการ Run Program แบบทีละคำสั่ง แต่จะไม่ลงไปถึงคำสั่งของ SubVI
-  Step Out ใช้สำหรับการ Run Program จนออกจาก Loop และ SubVI

2.2.3 Icon and Connector Pane

Icon and Connector Pane คือ เป็นรายละเอียดในส่วนของฟังก์ชัน หรือ SubVIs ต่างๆ ที่ใช้ในการเขียนหรือพัฒนาโปรแกรม รวมถึงรายละเอียดของการเชื่อมต่อหรือเชื่อมโยงข้อมูลระหว่าง VIs หรือ SubVIs เช่นเดียวกับการเขียนโปรแกรมแบบใช้ตัวอักษร ที่มีการติดต่อหรือเรียกใช้โปรแกรมย่อย (subroutine)

2.3 เครื่องมือต่างๆ ที่ใช้ในการออกแบบ

เครื่องมือที่ใช้ในการออกแบบ VIs จะสามารถแบ่งออกเป็น 3 ส่วนที่สำคัญๆ คือ

- Tools Palette
- Controls Palette
- Functions Palette

2.3.1 Tools Palette

Tools Palette คือ เครื่องมือที่ใช้ในการเขียนหรือพัฒนาโปรแกรมที่ใช้ทั้งในส่วนของการออกแบบใน Front Panel และในส่วนของการออกแบบใน Block Diagram



รูปที่ 2.3 แสดง Tools Palette

Tools Palette จะประกอบด้วยเครื่องมือต่าง ๆ ดังต่อไปนี้



Automatic Tool Selection ใช้สำหรับการเปลี่ยนเครื่องมือ (Tool) ต่างๆ โดยอัตโนมัติ



Operate Value ใช้สำหรับการเปลี่ยนแปลงค่าของ Controls และ Indicators




Position/Size/Select ใช้ในการเลือก (Select) หรือจัดวางตำแหน่ง (Position) ของอุปกรณ์ หรือการปรับขนาด (Size) ของ Control หรือ Indicator




Edit Text ใช้สำหรับการแก้ไขข้อความที่เป็นตัวอักษรหรือเพิ่มเติมข้อความลงบน Front Panel




Connect Wire ใช้สำหรับการเชื่อมโยงสายระหว่างวัตถุต่างๆ

 Object Shortcut Menu ใช้สำหรับแสดงและเลือกเมนูที่เกี่ยวข้องกับสิ่งต่างๆ บน Front Panel และ Block Diagram

 Scroll Window ใช้สำหรับการเลื่อน (Scroll Window) ทั้ง Front Panel และ Block Diagram

 Set/Clear Breakpoint ใช้สำหรับการกำหนดจุดหยุดการทำงานของโปรแกรมจากเงื่อนไขต่าง ๆ

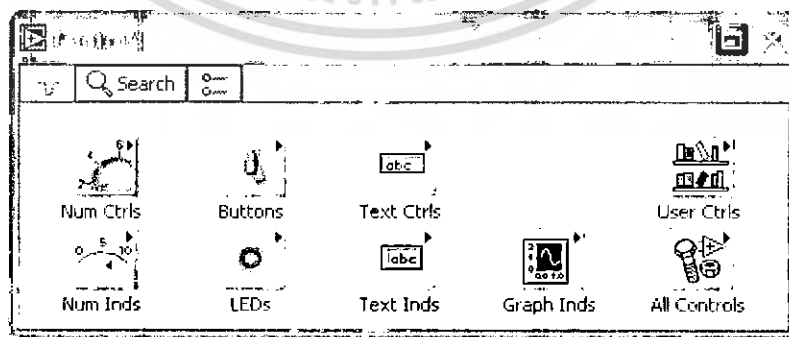
 Probe Data ใช้สำหรับกำหนดจุดที่ต้องการให้แสดงค่าของตัวแปรในการทำงานของโปรแกรม

 Get Color ใช้สำหรับการคัดลอก (Copy) สีที่เลือกจากวัตถุหนึ่งเพื่อที่จะนำไปใช้ในการเปลี่ยนสีของอีกวัตถุหนึ่งให้สีเหมือนกับวัตถุที่ Copy สีมา

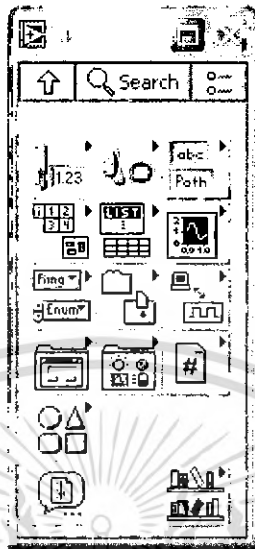
 Set Color ใช้สำหรับการเลือกสีให้กับอุปกรณ์ต่างๆ

2.3.2 Controls Palette

เครื่องมือที่ใช้ในการออกแบบโปรแกรมในส่วนของ Front Panel คือ Controls Palette โดยที่ LabVIEW จะมี Controls Palette เป็นส่วนที่ใช้ในการติดต่อกับผู้ใช้งาน โดยจะจัดเป็นกลุ่มต่าง ๆ เช่น กลุ่มของตัวเลข (Numeric) ซึ่งภายในกลุ่มจะมี Control และ Indicator ต่าง ๆ ที่เกี่ยวข้องกับตัวเลข รวมทั้งใช้ในการปรับเปลี่ยนค่าและแสดงผล เป็นต้น



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.4 แสดง Controls Palette

รายละเอียดของเครื่องมือหรือฟังก์ชันต่างๆ ในส่วนของ Controls Palette ของที่สำคัญ



Numeric เป็นส่วนของเครื่องมือที่ใช้ในการควบคุมหรือกำหนดค่า และการแสดงผล เช่น Numeric Control, Numeric Indicator, Knob, Gauge และ Thermometer เป็นต้น



Boolean เป็นส่วนของสวิตช์หรือปุ่มเปิดปิด รวมทั้ง LED แสดงผล ที่ใช้ในการกำหนดสถานะทางบูลีนลอจิก



String & Path เป็นส่วนของเครื่องมือที่ใช้ในการควบคุมหรือการกำหนดค่าที่เป็นข้อความ (String Control) และแสดงผล (String Indicator) และการกำหนดค่าเส้นทางในการเข้าถึงไฟล์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Array & Cluster เป็นส่วนของเครื่องมือที่ใช้ในการกำหนดค่าและแสดงผลในส่วนของตัวแปรที่เป็นอาร์เรย์



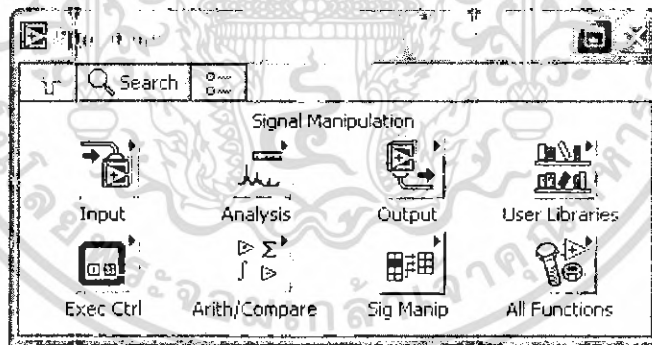
Graph เป็นส่วนของเครื่องมือที่ใช้ในการแสดงผลในรูปแบบของกราฟแบบต่างๆ



I/O เป็นส่วนของเครื่องมือที่ใช้ในการควบคุมและแสดงผลของการรับส่งข้อมูลในส่วนของ Input และ Output

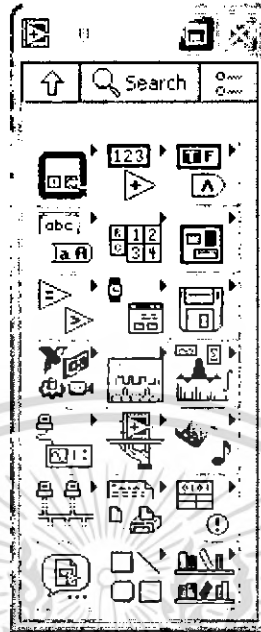
2.3.3 Functions Palette

LabVIEW จะใช้ Functions Palette เป็นเครื่องมือที่ใช้ในการออกแบบโปรแกรมในส่วนของ Block Diagram ซึ่งจะมี Function และ SubVIs ต่างๆ ที่มีอยู่แล้วให้ผู้ใช้ได้เลือกใช้งาน โดย Function และ SubVIs จะจัดเป็นกลุ่มๆ เช่น Numeric Function จะมี Function ต่างๆ เกี่ยวกับตัวเลข เช่น บวก ลบ คูณ หาร เป็นต้น



รูปที่ 2.5 แสดง Functions Palette

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.5 แสดง Functions Palette (ต่อ)

รายละเอียดของเครื่องมือหรือฟังก์ชันต่างๆ ในส่วนของ Functions Palette ของที่สำคัญ



Structures เป็นส่วนของฟังก์ชันเงื่อนไขต่างๆ ที่ใช้ในการเขียน โปรแกรม เช่น While Loop, For Loop และ Formula Node เป็นต้น



Numeric เป็นส่วนของฟังก์ชันในการดำเนินการทางคณิตศาสตร์, ตรีโกณมิติ และค่าคงที่ต่างๆ



Boolean เป็นส่วนของฟังก์ชันทางลอจิก เช่น And, Or, Not, True Constant และ False Constant เป็นต้น



String เป็นส่วนของฟังก์ชันต่างๆที่เกี่ยวข้องกับข้อความ เช่น String Constant และ String/Number Conversion เป็นต้น



Array เป็นส่วนของฟังก์ชันในการจัดการตัวแปรประเภทของตัวแปรอาร์เรย์



Comparison เป็นส่วนของฟังก์ชันและเงื่อนไขต่างๆ ที่ใช้ในการเปรียบเทียบ



Time & Dialog เป็นส่วนของฟังก์ชันทางเวลา ที่ใช้ในการกำหนดค่าของฐานเวลา



File I/O เป็นส่วนของฟังก์ชันที่ใช้ในการควบคุมและแสดงผลของการรับส่งข้อมูลในส่วนของ

Input และ Output



Analyze เป็นส่วนของฟังก์ชันที่ใช้ในการวิเคราะห์สัญญาณหรือข้อมูลต่างๆ



Instrument I/O เป็นส่วนของฟังก์ชันที่ใช้ในการควบคุมและแสดงผลของการรับส่งข้อมูลระหว่างคอมพิวเตอร์กับอุปกรณ์ภายนอก ที่มีการติดต่อสื่อสารผ่านทางพอร์ตอนุกรม, พอร์ตนาน หรือการ์ด DAQ เป็นต้น



Graphics & Sound เป็นส่วนของฟังก์ชันในการแสดงผลในรูปแบบของกราฟและเสียง



Communication เป็นส่วนของฟังก์ชันในการติดต่อสื่อสารกับอุปกรณ์ภายนอก



Select a VI... เป็นส่วนของการเลือกใช้ VIs ที่เป็นส่วนประกอบในการเขียนโปรแกรมหรือเลือกใช้ SubVIs ที่มีการสร้างไว้แล้ว

2.4 ประเภทของข้อมูล (Data Type)

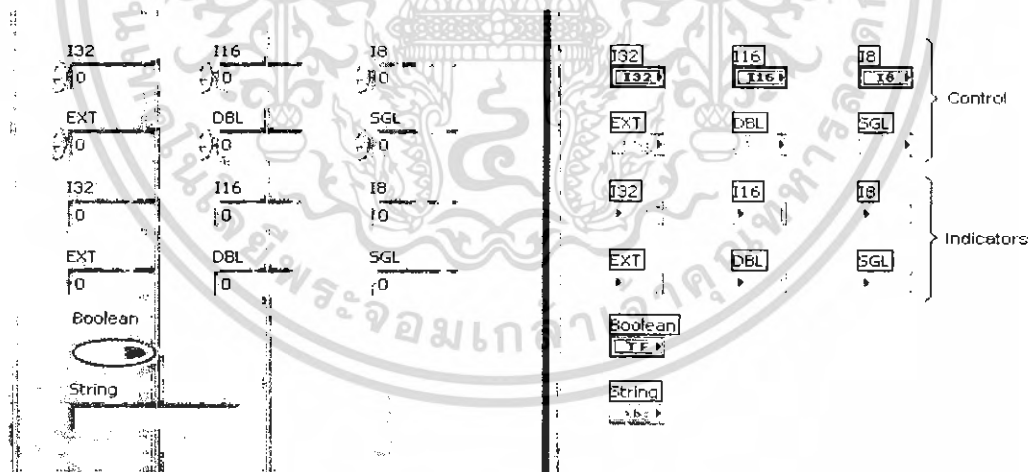
2.4.1 ประเภทของตัวแปร

เช่นเดียวกับในภาษาและโปรแกรมอื่นๆ เช่น ภาษา Basic, C หรือ Pascal ที่มีประเภทของข้อมูล โดยที่ประเภทของข้อมูลใน LabVIEW ที่สำคัญจะประกอบด้วย Numeric, Boolean, String และ Waveform เป็นต้น ซึ่งแต่ละรูปแบบของประเภทข้อมูลจะมีการแยกสีให้แตกต่างกัน (Color Code) และแต่ละประเภทของข้อมูลก็จะมีฟังก์ชันที่แยกตามประเภท รวมทั้งสีของสายเชื่อมโยง (Wiring) ก็จะมีขึ้นอยู่กับประเภทของข้อมูลนั้นๆ

ตารางที่ 2.1 แสดงรายละเอียดของตัวแปรต่างๆ (Numeric Control และ Indicators)

Control	Indicator	Data Type	Default Values
		Single-precision, floating-point numeric	0.0
		Double-precision, floating-point numeric	0.0
		Extended-precision, floating-point numeric	0.0
		Complex single-precision, floating-point numeric	0.0 + i0.0
		Complex double-precision, floating-point numeric	0.0 + i0.0
		Complex extended-precision, floating-point numeric	0.0 + i0.0
		8-bit signed integer numeric	0
		16-bit signed integer numeric	0
		32-bit signed integer numeric	0
		8-bit unsigned integer numeric	0
		16-bit unsigned integer numeric	0
		32-bit unsigned integer numeric	0

ประเภทของข้อมูลจะมีความสำคัญในการเขียนโปรแกรมใน LabVIEW เป็นอย่างมาก เพราะ ฟังก์ชัน หรือ SubVIs จะมีการรับข้อมูลในส่วนของ Input และให้ Output ที่แตกต่างกันออกไป



รูปที่ 2.6 แสดงตัวอย่างของตัวแปรต่างๆ ที่ใช้ใน LabVIEW

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.2 ประเภทของสายสัญญาณ

ประเภทของสายสัญญาณใน LabVIEW สามารถแบ่งออกได้เป็น 3 แบบ ดังต่อไปนี้ คือ

1. แบบ สเกลาร์
2. แบบ อาร์เรย์ 1 มิติ
3. แบบ อาร์เรย์ 2 มิติ



รูปที่ 2.7 แสดงประเภทของสายสัญญาณแบบต่างๆ ใน LabVIEW

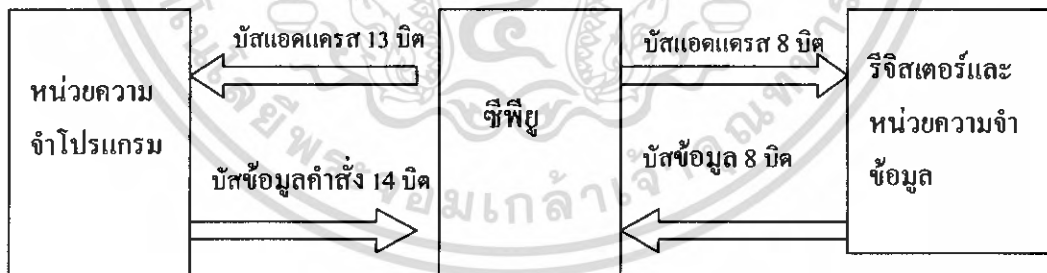
บทที่ 3

ทฤษฎีไมโครคอนโทรลเลอร์ PIC16F87x

3.1 โครงสร้างทางฮาร์ดแวร์ของไมโครคอนโทรลเลอร์ PIC16F87x

ไมโครคอนโทรลเลอร์ตระกูล PIC มีสถาปัตยกรรมแบบฮาร์วาร์ด (Harvard architecture) กล่าวคือ มีการแยกหน่วยความจำโปรแกรมและหน่วยความจำข้อมูลออกจากกัน โดยมีบัสสำหรับติดต่อแยกกันด้วย ดังแสดงในรูปที่ 3.1 จะเห็นว่าชิพภายในไมโครคอนโทรลเลอร์ จะติดต่อกับหน่วยความจำโปรแกรมด้วยบัสแอดเดรส 13 บิต และบัสข้อมูลหน่วยความจำโปรแกรม 14 บิต ในขณะที่บัสสำหรับติดต่อกับหน่วยความจำข้อมูลและรีจิสเตอร์ภายในเป็นแบบ 8 บิต ทั้งบัสแอดเดรสและบัสข้อมูล

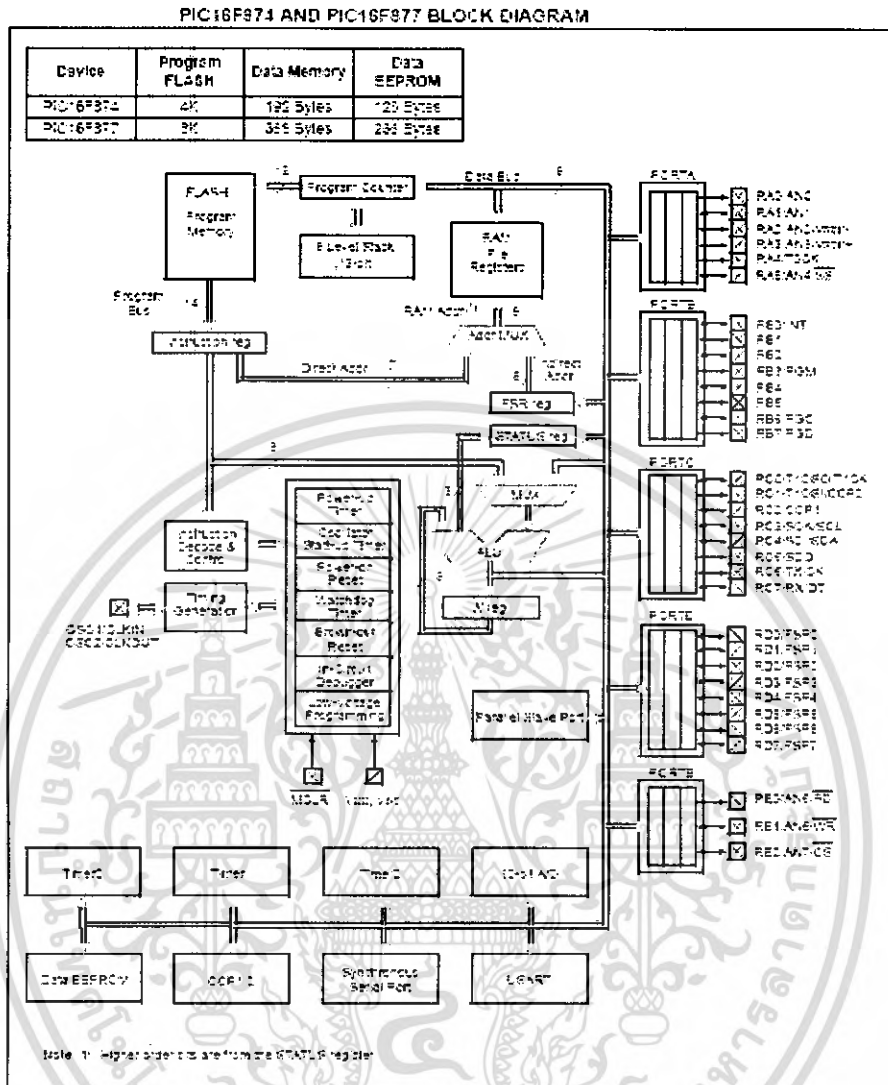
นอกจากการจัดสถาปัตยกรรมแบบนี้แล้ว การกระทำคำสั่งของไมโครคอนโทรลเลอร์ PIC ยังใช้กระบวนการที่เรียกว่า ไปป์ไลน์ (pipeline) ทำให้สามารถเฟตช์คำสั่งถัดไป ในขณะที่กำลังเอ็กซีคิวต์คำสั่งในปัจจุบัน ส่งผลให้ความเร็วในการทำงานของไมโครคอนโทรลเลอร์เพิ่มมากขึ้น นั่นจึงเป็นที่มาของความสามารถในการกระทำคำสั่ง 1 คำสั่งภายในสัญญาณนาฬิกา 1 ลูก (เฟตช์ : fetch เป็นกระบวนการเรียกคำสั่งออกจากหน่วยความจำโปรแกรมแล้วแปลเป็นเลขฐานสิบหกเพื่อให้ชิพเข้าใจส่วนกระบวนการเอ็กซีคิวต์ (execute) เป็นการกระทำคำสั่งให้เกิดผลลัพธ์ตามที่คำสั่งนั้นๆ กำหนด)



รูปที่ 3.1 โค้ดแอมแสดงรูปแบบสถาปัตยกรรมของไมโครคอนโทรลเลอร์แบบฮาร์วาร์ด

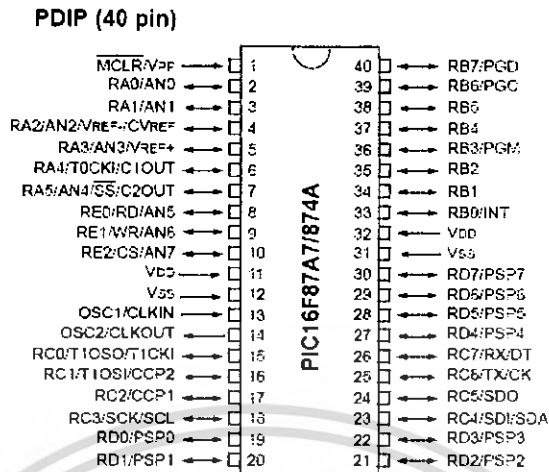
3.2 โครงสร้างของไมโครคอนโทรลเลอร์ PIC16F87x

แสดงดังในรูปที่ 3.2 ส่วนประกอบหลักก็จะเหมือนกับไมโครคอนโทรลเลอร์ PIC16F84 แต่จะมีส่วนที่เพิ่มเติมเข้ามาพอสมควร ได้แก่ วงจรบราวเอาต์รีเซต (brown – out reset), ส่วน



รูปที่ 3.2 (ข) โครงสร้างการทำงานของไมโครคอนโทรลเลอร์ PIC16F87x รุ่น 40 ขา เบอร์ PIC16F874 (A)/877(A)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.3 การจัดขาของไมโครคอนโทรลเลอร์ PIC16F87x

3.3 คุณสมบัติทางเทคนิคของ PIC16F87x/87xA

คุณสมบัติหลัก

- คำสั่ง
- 3.3.1 ซิพียูเป็นแบบ RISC (Reduced Instruction – Set Computer) มีคำสั่งใช้งานเพียง 35
 - 3.3.2 สามารถกระทำคำสั่งโดยใช้สัญญาณเพียงหนึ่งลูก ยกเว้นคำสั่งการกระโดด
 - 3.3.3 ความถี่สัญญาณนาฬิกา ตั้งแต่ไฟตรงถึง 20MHz
 - 3.3.4 หน่วยความจำโปรแกรม
 - 4 กิโลเวิร์ด สำหรับ PIC16F873(A)/874(A)
 - 8 กิโลเวิร์ด สำหรับ PIC16F876/876(A)/877A
 - 3.3.5 หน่วยความจำข้อมูลแรมหรือรีจิสเตอร์
 - 192 ไบต์ สำหรับ PIC16F873(A)/874(A)
 - 368 ไบต์ สำหรับ PIC16F(A)/877(A)
 - 3.3.6 ขนาดหน่วยความจำข้อมูลอีพรอม
 - 128 ไบต์ สำหรับ PIC16F873(A)/874(A)
 - 256 ไบต์ สำหรับ PIC16F876(A)/877(A)
 - 3.3.7 คอปสนองแหล่งกำเนิดอินเทอร์รัปต์สูงสุดถึง 15 แหล่งขึ้นกับเบอร์ของไมโครคอนโทรลเลอร์
 - 3.3.8 มีสแต็ก 8 ระดับ
 - 3.3.9 มีวงจรเพาเวอร์ออนรีเซต (POR)

3.3.10 มีเพาเวอร์อัปไทมเมอร์ (PWRT) และออสซิลเลเตอร์สตาร์ทอัปไทมเมอร์ (OST)

3.3.11 มีวงจรวอร์คโอ๊กไทมเมอร์ (WDT) ที่มีวงจรออสซิลเลเตอร์ในตัว ทำให้มีความน่าเชื่อถือในการทำงานสูง

3.3.12 เลือกป้องกันข้อมูลทั้งในหน่วยความจำโปรแกรมและหน่วยความจำข้อมูลสามารถเลือกระดับการป้องกันได้

3.3.13 มีโหมดประหยัดพลังงาน

3.3.14 สามารถโปรแกรมโดยใช้แรงดัน +5V ได้

3.3.15 แก้ไขข้อมูลในหน่วยความจำโปรแกรมด้วยกระบวนการ (In – Circuit Debugger) ผ่านพอร์ตเพียง 2 ขา

3.3.16 แก้ไขข้อมูลในหน่วยความจำโปรแกรมด้วยกระบวนการ ICD (In-Circuit Debugger) ผ่านพอร์ตเพียง 2 ขา

3.3.17 ซึ่พื้สามารถอ่านและเขียนหน่วยความจำโปรแกรมได้

3.3.18 ไฟเลี้ยง +2 ถึง +5.5V

3.3.19 กระแสซิงก์และซอร์สของพอร์ต 25mA

3.3.20 การใช้พลังงานไฟฟ้าในกรณีไม่ขับโหลด

น้อยกว่า 2mA ที่ไฟเลี้ยง +5V และสัญญาณนาฬิกา 4MHz

20 μA ที่ไฟเลี้ยง +3V และสัญญาณนาฬิกา 32kHz

น้อยกว่า 1 μA ในโหมดประหยัดพลังงานหรือสแตนด์บาย

คุณสมบัติพิเศษเพิ่มเติม

1. ไทมเมอร์ 3 ตัว คือ ไทมเมอร์ 0 ขนาด 8 บิต มีปริสเกลเลอร์ขนาด 8 บิตในตัว, ไทมเมอร์ 1 ขนาด 16 บิต พร้อมปริสเกลเลอร์ และไทมเมอร์ 2 ขนาด 8 บิต มีปริสเกลเลอร์, โพลด์สเกลเลอร์ และรีจิสเตอร์คาบเวลา (Period Register) ขนาด 8 บิตในตัว

2. มีโมดูล CCP 2 ชุด โดย

ส่วนตรวจจับสัญญาณหรือแคปเจอร์ (Capture) มีขนาด 16 บิต ความละเอียดสูงสุด 12.5 นาโนวินาที

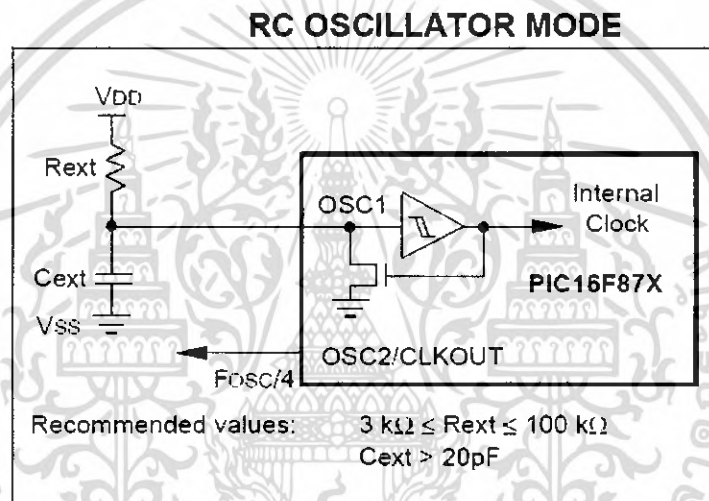
ส่วนเปรียบเทียบสัญญาณ (Compare) มีขนาด 16 บิต ความละเอียดสูงสุด 200 นาโนวินาที วงจร PWM มีความละเอียดสูงสุด 10 บิต

3. มีวงจรแปลงสัญญาณอะนาลอกเป็นดิจิตอลขนาด 10 บิต (5 ช่องสำหรับ PIC16F873(A)/876(A) และ 8 ช่องสำหรับ PIC16F874(A)/877(A))

4. วงจรเชื่อมต่ออุปกรณ์อนุกรมทั้ง SPI และบัส I²C
5. วงจรสื่อสารข้อมูลอนุกรม (USART) พร้อมการตรวจนับแอดเดรส 9 บิต
6. มีวงจรตรวจนับระดับแรงดันไฟเลี้ยง (บราวเอาต์ดีเทกชัน : Brown-out detection) เพื่อการรีเซ็ตซีพียู หรือเรียกว่า บราวเอาต์รีเซ็ต (Brown-out reset : BOR)

3.4 โหมดสัญญาณนาฬิกา

PIC16F87x สามารถเลือกโหมดของสัญญาณนาฬิกาเพื่อกำหนดจังหวะการทำงานได้มากถึง 4 โหมด โดยการกำหนดที่บิต FOSCO และ FOSC1 ในรีจิสเตอร์ Configuration Word ซึ่งในการทำงานจะต้องเลือกโหมดใดโหมดหนึ่ง ดังมีรายละเอียดต่อไปนี้



รูปที่ 3.4 การต่อตัวต้านทานและตัวเก็บประจุเพื่อกำหนดความถี่สัญญาณนาฬิกาเมื่อไมโครคอนโทรลเลอร์ PIC16F87X ทำงานในโหมด RC

1. โหมด LP ใช้กับคริสตอลหรือเซรามิกเรโซเนเตอร์พลังงานต่ำความถี่ 32kHz-200kHz
2. โหมด XT ใช้กับคริสตอลหรือเซรามิกเรโซเนเตอร์มาตรฐานความถี่ 200kHz-4MHz
3. โหมด HS ใช้กับคริสตอลหรือเซรามิกเรโซเนเตอร์ความถี่สูง 4MHz - 20MHz
4. โหมด RC (External Resistor/Capacitor) สามารถกำหนดค่าความถี่ได้จากค่าของตัวต้านทานและตัวเก็บประจุที่ต่อภายนอกเข้ากับขา OSC1/CLKIN ดังแสดงในรูปที่ 3.4 ความถี่สูงสุดคือ 4MHz อย่างไรก็ตามความถี่ของสัญญาณนาฬิกาในโหมดนี้ไม่อาจกำหนดลงไปได้อย่างชัดเจน เนื่องจากต้องพิจารณาถึงองค์ประกอบที่สามารถเปลี่ยนแปลงได้ในขอบเขตที่กว้าง ไม่ว่าจะเป็นค่าของแรงดันไฟเลี้ยง, ค่าของต้านและตัวเก็บประจุ ซึ่งต้องรวมไปถึงค่าความผิดพลาดของ

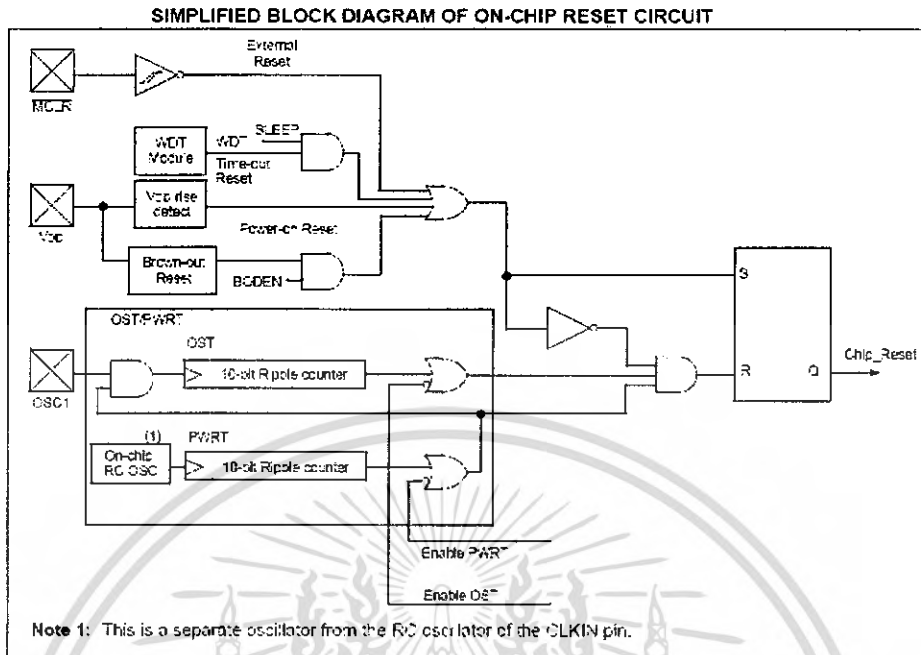
อุปกรณ์ทั้งสองด้วย อย่างไรก็ตามค่าของตัวต้านทานที่เหมาะสมอยู่ในย่าน $3k\Omega - 100k\Omega$ ส่วนค่าของตัวเก็บประจุควรมากกว่า $20pF$ นอกจากนี้ที่ขา OSC2/CLKOUT จะมีสัญญาณนาฬิกาความถี่ $\frac{1}{4}$ เท่าของความถี่สัญญาณนาฬิกาหลักส่งออกมา

3.5 กระบวนการรีเซ็ตใน PIC16F87x

การรีเซ็ต (reset) เป็นกระบวนการที่สำคัญมากในระบบไมโครคอนโทรลเลอร์ เป็นการกำหนดให้ซีพียูภายในไมโครคอนโทรลเลอร์เริ่มต้นการทำงานใหม่ ทั้งนี้เพื่อประโยชน์ในการแก้ไขความผิดปกติหรือการทำงานที่ผิดพลาดของไมโครคอนโทรลเลอร์ อันทำให้ไมโครคอนโทรลเลอร์ทำงานค้างอยู่ในสถานะใดสถานะหนึ่งหรือหยุดทำงาน เมื่อทำการรีเซ็ตไมโครคอนโทรลเลอร์ก็จะกลับมาเริ่มต้นการทำงานใหม่ ทำให้ระบบโดยรวมยังคงกลับมาทำงานได้ต่อไป การรีเซ็ตในไมโครคอนโทรลเลอร์ PIC16F87x มีด้วยกัน 6 ประเภท ดังนี้

1. เพาเวอร์ออนรีเซ็ต (power – on reset) เป็นการรีเซ็ตที่เกิดขึ้นหลังจากเริ่มต้นจ่ายไฟเลี้ยงใหม่ให้แก่ไมโครคอนโทรลเลอร์
2. การรีเซ็ตที่ขา \overline{MCLR} ในระหว่างการทำงานปกติ
3. การรีเซ็ตที่ขา \overline{MCLR} ขณะทำงานในโหมดสลีป
4. การรีเซ็ตจากวอตช์ดีดอทไทมเมอร์ ในขณะที่ทำงานปกติ
5. วอตช์ดีดอทไทมเมอร์เวกอัปขณะทำงานในโหมดสลีป
6. การรีเซ็ตเนื่องจากไฟเลี้ยงลดต่ำลงจากที่กำหนดโดยวงจรบราวเอาต์ดีเท็ก (BOD : Brown – Out Detect) เรียกว่า บราวเอาต์รีเซ็ต (brown – out reset)

ในรูปที่ 3.5 แสดงไดอะแกรมของการเกิดสัญญาณรีเซ็ตในไมโครคอนโทรลเลอร์ PIC16F87x



รูปที่ 3.5 แสดงกลไกของการเกิดรีเซ็ตในไมโครคอนโทรลเลอร์ PIC16F87X

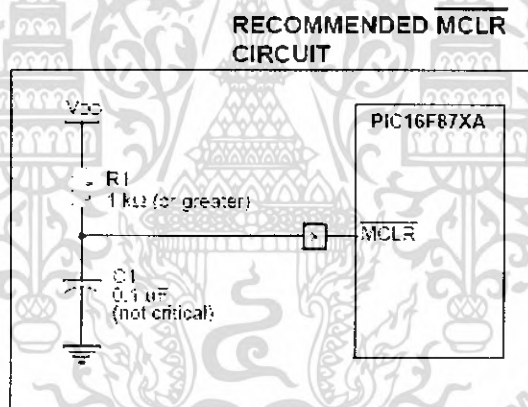
POR	BOR	TO	PD	สิ่งที่เกิดขึ้นในระบบ
0	X	1	1	เพาเวอร์ออนรีเซ็ต
0	X	0	X	ผิดพลาด เพราะบิต \overline{TO} ต้องเซตในภาวะเพาเวอร์ออนรีเซ็ต
0	X	X	0	ผิดพลาด เพราะบิต \overline{PD} ต้องเซตในภาวะเพาเวอร์ออนรีเซ็ต
1	0	1	1	วงจรบราวเอาต์ดีเท็กทำงาน เกิดรีเซ็ต
1	1	0	1	เกิดการรีเซ็ตเนื่องจาก WDT ไทม์เอาต์ในโหมดปกติ
1	1	0	0	เวกอัพเนื่องจาก WDT ไทม์เอาต์ในโหมดสลีป
1	1	U	U	เกิดการรีเซ็ตที่ \overline{MCLR} ในโหมดปกติ
1	1	1	0	เกิดการรีเซ็ตที่ \overline{MCLR} ในโหมดสลีป

ตารางที่ 3.1 แสดงความสัมพันธ์ระหว่างบิตแสดงสถานะกับลักษณะการเกิดรีเซ็ตขึ้นของไมโครคอนโทรลเลอร์ PIC16F87x

การรีเซ็ตในแต่ละประเภทจะส่งผลกระทบต่อรีจิสเตอร์ที่แตกต่างกัน รีจิสเตอร์ส่วนใหญ่จะเข้าสู่สภาวะรีเซ็ต (Reset State) เมื่อเกิดเพาเวอร์อนรีเซ็ต, การรีเซ็ตที่ขา $\overline{\text{MCLR}}$, การรีเซ็ตเนื่องจากวอตช์ดีค็อกไทมเมอร์ และการรีเซ็ตเนื่องจากบราวเอาต์คัสเท็ก และจะไม่ได้รับผลกระทบในกรณีที่วอตช์ดีค็อกไทมเมอร์เวกอัป ผู้ใช้งานสามารถตรวจสอบสาเหตุของการรีเซ็ตได้จากบิตแสดงสถานะ 4 บิต คือ บิต $\overline{\text{TO}}$ กับ $\overline{\text{PD}}$ ในรีจิสเตอร์ STATUS และบิต POR กับ $\overline{\text{BOR}}$ ในรีจิสเตอร์ PCON ดังสรุปไว้แล้วในตารางที่ 3.1

3.6 การรีเซ็ตที่ขา $\overline{\text{MCLR}}$

เกิดขึ้นเมื่อขา $\overline{\text{MCLR}}$ ได้รับลอจิก “0” ในการใช้งานปกติ ขานี้จะต่อตัวต้านทานพูลอัปหรือวงจรเพาเวอร์อนรีเซ็ต ซึ่งหลังจากจ่ายไฟแล้ว สถานะที่ขา $\overline{\text{MCLR}}$ จะถูกรักษาไว้ที่ระดับลอจิก “1” ในกรณีที่ต้องการให้เกิดการรีเซ็ตสามารถต่อขา $\overline{\text{MCLR}}$ นี้ลงกราวด์ชั่วคราว แล้วปลดออก จะทำให้เกิดการรีเซ็ตแล้ว ดังนั้นในทางปฏิบัติอาจต่อสวิตช์กดติดปล่อยดับเข้าระหว่างขา $\overline{\text{MCLR}}$ นี้กับกราวด์ หากต้องการให้เกิดการรีเซ็ต ก็เพียงกดสวิตช์แล้วปล่อยเท่านั้น



รูปที่ 3.6 วงจรเพาเวอร์อนรีเซ็ตสำหรับไมโครคอนโทรลเลอร์ PIC16F87x

การรีเซ็ตแบบนี้จะทำให้ไมโครคอนโทรลเลอร์กลับมาเริ่มทำงานที่แอดเดรส 0x0000 ใหม่ค่าในรีจิสเตอร์บางตัวไม่เปลี่ยนแปลง และข้อมูลในหน่วยความจำข้อมูลแรกจะไม่ได้รับผลกระทบถ้าหากไมโครคอนโทรลเลอร์ทำงานในโหมดสลีป และมีการป้อนลอจิก “0” เข้าที่ขา $\overline{\text{MCLR}}$ นอกจากทำให้เกิดการรีเซ็ตแล้วยังทำให้เกิดการเวกอัปออกจากโหมดสลีปด้วย

3.7 พอร์ตอินพุตเอาต์พุตของไมโครคอนโทรลเลอร์ PIC16F87x

PIC16F87x มีพอร์ตให้ใช้งานตั้งแต่ 3-5 พอร์ต จำนวน 20-33 บิต ขึ้นอยู่กับเบอร์ของไมโครคอนโทรลเลอร์ ดังอธิบายไปแล้วในบทที่ 2 และด้วยความสามารถของพอร์ตใน PIC16F87x ที่สามารถทำงานได้หลายอย่าง จึงจำเป็นอย่างยี่งที่ผู้ใช้งานต้องทำความเข้าใจถึงโครงสร้างทางฮาร์ดแวร์และการกำหนดหรือเลือกฟังก์ชันการทำงานให้แก่ขาพอร์ตแต่ละขาด้วยกระบวนการทางซอฟต์แวร์ ทั้งนี้เพื่อให้สามารถใช้งานพอร์ตทั้งหมดของ PIC16F87x ได้อย่างมีประสิทธิภาพสูงสุด

ในบทนี้จะกล่าวถึงภาพรวมของพอร์ตทั้งหมด ตั้งแต่พอร์ต A ถึง E โดยจะเน้นไปที่โครงสร้างทางฮาร์ดแวร์และฟังก์ชันการทำงานในภาพรวม สำหรับหน้าที่หรือฟังก์ชันพิเศษที่ขาพอร์ตนั้นๆ สามารถทำได้จะกล่าวถึงในรายละเอียดต่อไป

ความสามารถในการจ่ายกระแสเอาต์พุตของขาพอร์ตที่ไฟเลี้ยง +5V คือ 25mA ต่อขาทั้งกระแสซิงค์และกระแสซอร์ซ ในขณะที่กระแสเอาต์พุตรวมของพอร์ต A, B และ E มีค่าสูงสุด 200mA ส่วนกระแสเอาต์พุตรวมของพอร์ต C และ D มีค่าสูงสุด 200mA ดังนั้นในการออกแบบเพื่อขับโหลดทางเอาต์พุตของขาพอร์ตต้องระวังเรื่องกระแสเอาต์พุตรวมที่ไมโครคอนโทรลเลอร์สามารถขับได้ด้วย

3.7.1 พอร์ต A

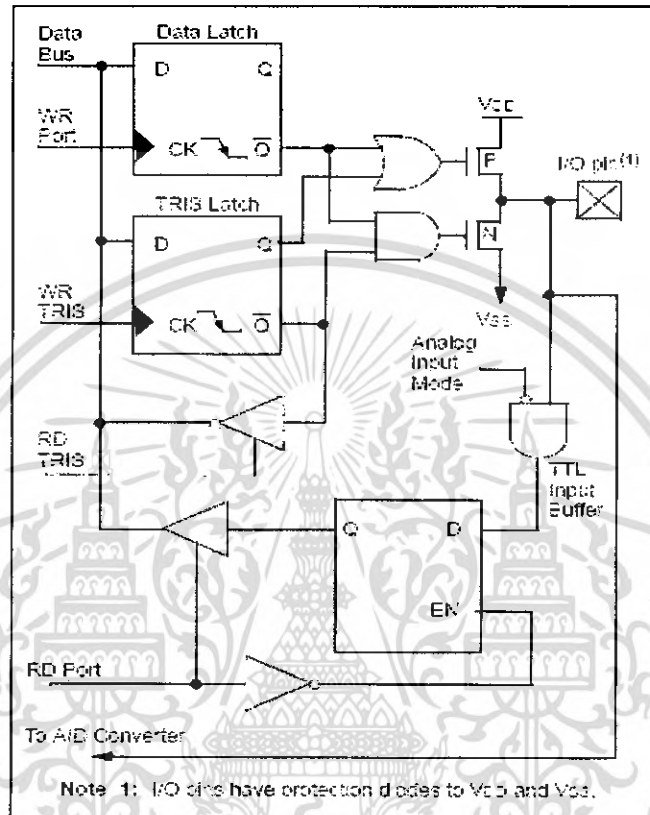
มีทั้งสิ้น 6 ช่องหรือ 6 บิต กำหนดชื่อขาเป็น RA0-RA5 รีจิสเตอร์ที่ใช้ในการเก็บข้อมูลคือ PORTA มีแอดเดรสอยู่ที่ 0x05 (แแบงก์ 0) เป็นรีจิสเตอร์ขนาด 8 บิต แต่ใช้งานจริงเพียง 6 บิต ที่เหลือ 2 บิต ต้องกำหนดให้เป็น "0" ส่วนการกำหนดทิศทางของพอร์ตนี้กระทำผ่านรีจิสเตอร์ TRISA ซึ่งมีแอดเดรสอยู่ที่ 0x85 (แแบงก์ 1) มีขนาด 8 บิตและใช้เพียง 6 บิต เช่นกัน 2 บิตบนคือบิต 6 และบิต 7 ต้องกำหนดให้เป็น "0" บิต 0 ของ TRISA ใช้กำหนดทิศทางของขาพอร์ต RA0 ไล่เรียงลำดับจนถึงบิต 5 ของ TRISA ใช้กำหนดทิศทางของขาพอร์ต RA5 หากต้องการกำหนดให้ขาพอร์ตในบิตใดเป็นอินพุตต้องเขียนข้อมูล "1" ไปยังบิตนั้น และในทางตรงข้ามหากต้องการกำหนดให้เป็นขาเอาต์พุตให้เขียนข้อมูล "0" ไปยังบิตนั้น

โครงสร้างทางฮาร์ดแวร์

พอร์ต A สามารถทำงานเป็นขาพอร์ตอินพุตเอาต์พุตปกติและเป็นขาอินพุตสัญญาณอะนาล็อกสำหรับวงจรแปลงสัญญาณอะนาล็อกเป็นดิจิตอลขนาด 10 บิต ภายในไมโครคอนโทรลเลอร์ โดยขา RA0-RA3 และ RA5 จะมีการทำงานที่เหมือนกัน ส่วน RA4 จะแตกต่างตรงที่ขานี้

นอกจากเป็นขาพอร์ตอินพุตเอาต์พุตปกติแล้ว ยังใช้เป็นขาอินพุตสำหรับไทมเมอร์ 0 ภายในไมโครคอนโทรลเลอร์ด้วย และขา RA4 นี้ไม่สามารถใช้งานขาอินพุตรับสัญญาณอะนาลอกได้

**BLOCK DIAGRAM OF
RA3:RA0 AND RA5 PINS**



รูปที่ 3.7 โครงสร้างขา RA0-RA3 ของพอร์ต A ในไมโครคอนโทรลเลอร์ PIC16F877x

ในรูปที่ 3.7 แสดงโคแอสแกรมของพอร์ต A ในบิต RA0-RA3 จะเห็นได้ว่าที่ขาพอร์ตจะมีแอนด์เกตทำหน้าที่เลือกลักษณะการทำงานของขาพอร์ตเมื่อเป็นขาอินพุต หากเลือกให้ขาพอร์ตนี้เป็นขาพอร์ตอินพุตดิจิทัล สัญญาณเลือกโหมดอินพุตอะนาลอกจะต้องเป็นลอจิก "0" แต่ถ้าหากต้องการกำหนดให้เป็นขาอินพุตสัญญาณอะนาลอก สัญญาณเลือกโหมดต้องเป็นลอจิก "1" สัญญาณอะนาลอกที่ป้อนเข้ามายังขา นี้ จะผ่านเข้าสู่วงจรแปลงสัญญาณอะนาลอกเป็นดิจิทัลหรือวงจรเปรียบเทียบแรงดันอะนาลอก (เฉพาะในอนุกรม PIC16F87xA) โดยตรง

เมื่อขาพอร์ต RA0-RA3 ทำงานเป็นขาพอร์ตอินพุตดิจิทัล จะสามารถรับสัญญาณดิจิทัลระดับที่ทีแอล (0-5V) ได้โดยตรง หากทำงานเป็นเอาต์พุตจะสามารถขับโหลดที่ต้องการ

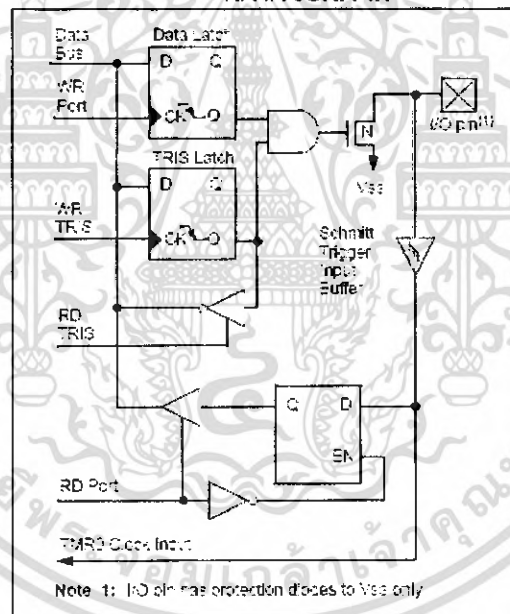
กระแส 20mA ได้หาคำนามาขับ LED ต้องต่อตัวต้านทานจำกัดกระแส หรือถ้าใช้ไฟเลี้ยง 3V ก็ จะขับ LED ได้โดยตรง

ในรูปที่ 3.8 เป็นไออะแกรมของขาพอร์ต RA4/T0CKI โดยขานี้จะเป็นขาพอร์ตอินพุต เอาต์พุตปกติ และขาอินพุตรับสัญญาณนาฬิกาจากภายนอกของโมดูลไทมเมอร์ 0 วงจรอินพุต บัพเฟอร์ที่ขาพอร์ตนี้เป็นแบบขมิตต์ทริกเกอร์ ทั้งนี้เพื่อจัดการให้สัญญาณอินพุตที่เข้ามามีความ สมบูรณ์มากที่สุด และจะต้องต่อตัวต้านทานพูลอัปค่าประมาณ 4.7k-10k ที่ขานี้เสมอเมื่อใช้งาน เป็นอินพุต

สำหรับใน PIC16F87xA ขาพอร์ต RA4 ยังใช้เป็นขาเอาต์พุตของโมดูลเปรียบเทียบ แรงดันอะนาลอกชุดที่ 1 (CIOUT) ด้วย โครงสร้างภายในจึงมีการเปลี่ยนแปลงจากเดิมไป เล็กน้อย ดังแสดงในรูปที่ 3.8 (ข)

ขา RA4 ของ PIC16F87X มาตรฐาน

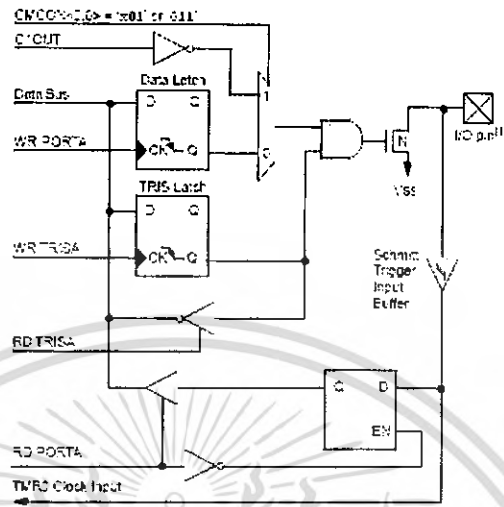
BLOCK DIAGRAM OF RA4/T0CKI PIN



รูปที่ 3.8 (ก)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขา RA4 ของ PIC16F87XA
BLOCK DIAGRAM OF RA4/T0CKI PIN



Note 1: IO pin free protector closes to V_{SS} only

รูปที่ 3.8 (ข)

รูปที่ 3.8 โครงสร้างขา RA4 ของพอร์ต A ในไมโครคอนโทรลเลอร์ PIC16F87x

ในกรณีที่ใช้งานเป็นพอร์ตเอาต์พุต วงจรเอาต์พุตจะเป็นแบบแคเรนเปิด (Open drain) ถ้าหากเทียบกับวงจรทรานซิสเตอร์ก็คือ วงจรคอลเล็กเตอร์เปิด (Open collector) นั่นเอง ในการใช้งานจึงต้องตัวต้านทานอนุกรมกับโหลดและไฟเลี้ยงของโหลด ดังวงจรในรูปที่ 3.8 โดยปกติถ้าหากไม่มีความจำเป็นควรกำหนดให้ขา RA4/T0CKI นี้เป็นอินพุตจะดีที่สุด โดยต้องต่อตัวต้านทานพูลอัพเสมอ หรือถ้าหากจำเป็นต้องใช้เป็นเอาต์พุตก็ต้องไม่ลืมว่า ขานี้เป็นเอาต์พุตแบบแคเรนเปิด ต้องต่อขั้วโหลดในแบบกระแสซิงก์เท่านั้น

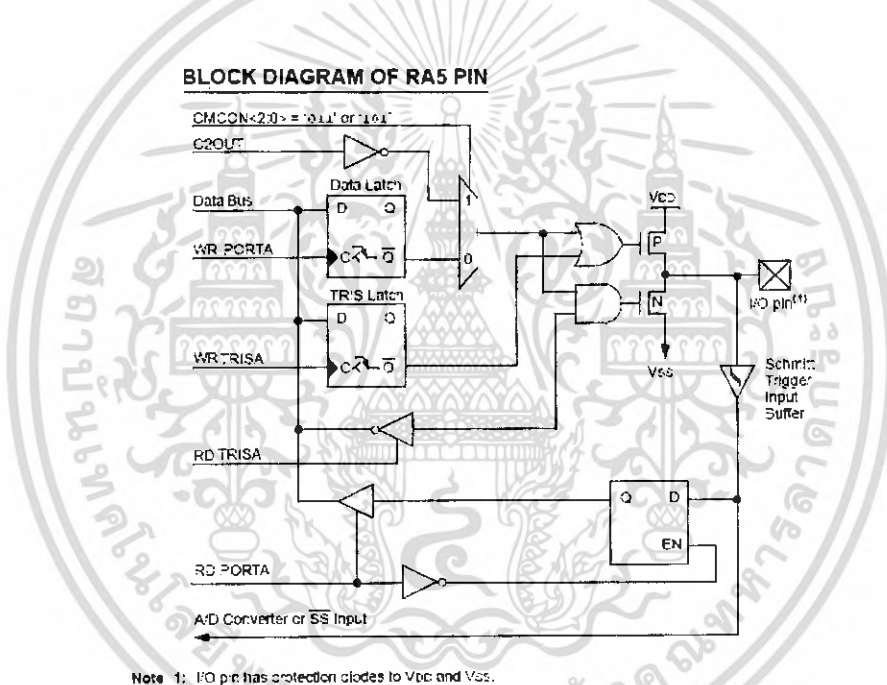
สำหรับในอนุกรม PIC16F87xA ขาพอร์ต RA4 ยังใช้เป็นขาเอาต์พุตของวงจรเปรียบเทียบแรงดันอะนาล็อกชุดที่ 1 ด้วย โดยโครงสร้างของเอาต์พุตนี้ก็ยังคงเป็นแบบแคเรนเปิด เช่นเดียวกับการใช้ขา RA4 เป็นพอร์ตดิจิทัลเอาต์พุตปกติ

ในรูปที่ 3.9 แสดงโครงสร้างของพอร์ต RA5 ขานี้ก็มีความพิเศษไม่น้อย โดยนอกจากเป็นขาพอร์ตอินพุตเอาต์พุตดิจิทัลแล้วยังสามารถเลือกให้ทำงานเป็นขาพอร์ตอินพุตสัญญาณอะนาล็อกสำหรับวงจรแปลงสัญญาณอะนาล็อกเป็นดิจิทัลในช่อง 4 และเป็นขาเอาต์พุตของวงจรเปรียบเทียบแรงดันอะนาล็อกชุดที่ 2 ในไมโครคอนโทรลเลอร์อนุกรม PIC16F87xA นอกจากนั้นยังใช้เป็นขา Slave Select (SS) ในกรณีที่เลือกใช้การสื่อสารข้อมูลอนุกรมซิงโครนัสในแบบ SPI ซึ่งจะได้กล่าวถึงในรายละเอียดต่อไป

การติดต่อเพื่อกำหนดการทำงานและเขียนข้อมูลไปยังพอร์ต A

ในขั้นแรกต้องทำการเตรียมความพร้อมให้แก่รีจิสเตอร์ PORTA โดยการเลือกแบงก์สำหรับติดต่อกับรีจิสเตอร์ PORTA จากนั้นส่งข้อมูล "0" เพื่อเคลียร์ข้อมูลทั้งหมด และเนื่องจากพอร์ต A ทำงานกับสัญญาณอะนาลอกได้ และค่าเริ่มต้นของพอร์ต A ที่ทำงานกับสัญญาณอะนาลอกจะถูกกำหนดให้เป็นอินพุตอะนาลอกทั้งหมด ดังนั้นหากต้องการใช้งานเป็นพอร์ตดิจิทัลต้องกำหนดข้อมูล 0x06 หรือ 0x07 แล้วเขียนลงในรีจิสเตอร์ ADCON1 เพื่อคิเสอเบิลการทำงานกับสัญญาณอะนาลอก

จากนั้นจึงเลือกแบงก์ใหม่เพื่อติดต่อกับรีจิสเตอร์ TRISA แล้วเขียนข้อมูลเพื่อกำหนดทิศทางของขาพอร์ตตามที่ต้องการลงในรีจิสเตอร์ TRISA



โครงสร้างขาพอร์ต RA5 ของพอร์ต A ในไมโครคอนโทรลเลอร์ PIC16F87X

รูปที่ 3.9 โครงสร้างขาพอร์ต RA5 ของพอร์ต A ใน ไมโครคอนโทรลเลอร์ PIC16F87X

3.7.2 พอร์ต B

มี 8 บิต กำหนดชื่อขาเป็น RB0-RB7 รีจิสเตอร์ที่ใช้ในการเก็บข้อมูลคือ PORTB มีแอดเดรสอยู่ที่ 0x06 (แบงก์ 0) และ 0x106 (แบงก์ 2) เป็นรีจิสเตอร์ขนาด 8 บิต ส่วนการกำหนดทิศทางของพอร์ตนี้กระทำผ่านรีจิสเตอร์ TRISB ซึ่งมีแอดเดรสอยู่ที่ 0x86 (แบงก์ 1) และ 0x186 (แบงก์ 3) มีขนาด 8 บิต เช่นเดียวกับพอร์ต A บิต 0 ของ TRISB ใช้กำหนด

ทิศทางของขาพอร์ต RBO ไต่เรียงลำดับจนถึงบิต 7 ของ TRISB ใช้กำหนดทิศทางของขาพอร์ต RB7 หากต้องการกำหนดให้ขาพอร์ตในบิตใดเป็นอินพุตต้องเขียนข้อมูล “1” ไปยังบิตนั้น ในทางตรงข้ามหากต้องการกำหนดให้เป็นขาเอาต์พุตให้เขียนข้อมูล “0” ไปยังบิตนั้น

โครงสร้างทางฮาร์ดแวร์

พอร์ต B สามารถใช้งานในลักษณะต่างๆ ได้ 5 แบบคือ

1. เป็นขาพอร์ตอินพุตเอาต์พุตปกติ
2. เป็นขาอินพุตสัญญาณอินเทอร์รัปต์จากภายนอก โดยใช้ขา RBO/\overline{INT}
3. เป็นขาพอร์ตอินพุตสำหรับรับแรงดันโปรแกรมระดับต่ำ (low voltage programming)

โดยใช้ขา RB3/PGM

4. เป็นขาข้อมูลอนุกรมและสัญญาณนาฬิกาอนุกรมสำหรับการโปรแกรมหน่วยความจำ ภายในไมโครคอนโทรลเลอร์ ซึ่งใช้ 2 ขา คือ RB7/PGD และ RB6/PGC

5. ใช้เป็นแหล่งกำเนิดสัญญาณอินเทอร์รัปต์แบบตรวจสอบการเปลี่ยนแปลงข้อมูลหรือ ระดับสัญญาณที่ขา RB4-RB7

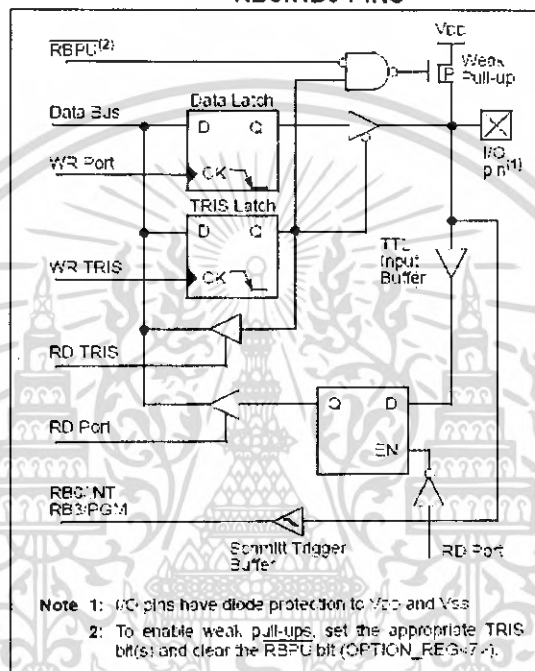
ในรูปที่ 3.10 แสดงไดอะแกรมของพอร์ต B ในบิต RB0-RB3 จะเห็นได้ว่าที่ขาพอร์ต จะมีวงจรพูลอัพแบบโปรแกรมได้ค่ออยู่ นั่นคือหากต้องการกำหนดให้เป็นขาอินพุต ต้องทำการเขียนข้อมูล “0” ไปยังบิต RBPU ในรีจิสเตอร์ OPTION_REG เพื่อเอ็นเอเบิลวงจรพูลอัพภายในขาพอร์ต B ในขณะที่หากกำหนดเป็นเอาต์พุต การพูลอัพที่ขาพอร์ต B นี้จะถูกยกเลิกโดยอัตโนมัติ นอกจากนั้นการพูลอัพนี้จะได้รับการยกเลิกเมื่อเกิดเพาเวอร์อนรีเซตขึ้น ในกรณีใช้ขา RBO/\overline{INT} เพื่อรับสัญญาณอินเทอร์รัปต์จากภายนอก สัญญาณจะผ่านเข้าไปยังวงจรบัฟเฟอร์แบบซิมิลาร์ทริกเกอร์เพื่อให้สัญญาณที่ได้มีความแม่นยำและมีเสถียรภาพ

ในรูปที่ 3.10 (ข) เป็นไดอะแกรมของขาพอร์ต RB4-RB7 โดยขาพอร์ตในกลุ่มนี้มีความสามารถพิเศษพอสมควร โดยสามารถเลือกให้ทำงานเป็นขาพอร์ตอินพุตเอาต์พุตปกติ, ขาอินพุตรับแรงดันสำหรับการโปรแกรม (RB3), ขาสัญญาณสำหรับการโปรแกรม (RB6-RB7) และทำงานเป็นแหล่งกำเนิดสัญญาณอินเทอร์รัปต์ในแบบตรวจสอบการเปลี่ยนแปลงที่ขาพอร์ต RB4-RB7

วงจรอินพุตบัฟเฟอร์ที่ขาพอร์ตนี้มีทั้งแบบทีทีแอลและซิมิลาร์ทริกเกอร์ ทั้งนี้เพื่อจัดการให้สัญญาณอินพุตที่เข้ามามีความเหมาะสมและสมบูรณ์มากที่สุด และยังคงสามารถรองรับการพูลอัพภายในแบบอัตโนมัติได้

ในกรณีที่มีการเขียนเปิดการตอบสนองอินเทอร์รัปต์แบบตรวจสอบการเปลี่ยนแปลงลอจิกที่พอร์ต RB4-RB7 หากเกิดการอินเทอร์รัปต์ขึ้น บิต RBIF (บิต 0 ในรีจิสเตอร์ INTCON) จะเซตและหลังจากตอบสนองการอินเทอร์รัปต์แล้ว ต้องเคลียร์บิต RBIF ด้วยกระบวนการทางซอฟต์แวร์เสมอ

**BLOCK DIAGRAM OF
RB3:RB0 PINS**

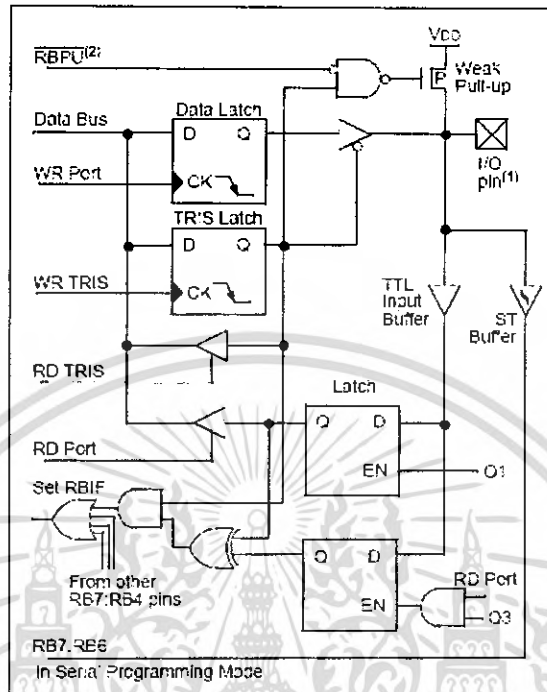


(ก) ขา RB0-RB3

รูปที่ 3.10 (ก)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

BLOCK DIAGRAM OF RB7:RB4 PINS



(ข) ขา RB4-RB7

รูปที่ 3.10 โครงสร้างของขาพอร์ต B ในไมโครคอนโทรลเลอร์ PIC16F87x

3.7.3 พอร์ต C

มีทั้งสิ้น 8 บิต กำหนดชื่อขาเป็น RC0-RC7 รีจิสเตอร์ที่ใช้เก็บข้อมูลคือ PORTC มีแอดเดรสอยู่ที่ 0x07 (แบงก์ 0) เป็นรีจิสเตอร์ขนาด 8 บิต ส่วนการกำหนดทิศทางของพอร์ตนี้กระทำผ่านรีจิสเตอร์ TRISC มีแอดเดรสอยู่ที่ 0x87 (แบงก์ 1) มีขนาด 8 บิต เช่นเดียวกับพอร์ต A และ B บิต 0 ของ TRISC ใช้กำหนดทิศทางของขาพอร์ต RC0 ได้เรียงลำดับจนถึงบิต 7 ของ TRISC ใช้กำหนดทิศทางของขาพอร์ต RC7 หากต้องการกำหนดให้ขาพอร์ตในบิตใดเป็นอินพุตต้องเขียนข้อมูล "1" ไปยังบิตนั้น และในทางตรงข้ามหากต้องการกำหนดให้เป็นขาเอาต์พุต ให้เขียนข้อมูล "0" ไปยังบิตนั้น

ไดอะแกรมของพอร์ต C ในบิต RC0-RC2 และ RC5-RC7 จะเห็นได้ว่ามีสัญญาณควบคุมการทำงานของขาพอร์ตมากมาย ทั้งนี้เนื่องจากขาพอร์ต C สามารถทำงานได้หลากหลายนั่นเอง สัญญาณควบคุมที่สำคัญคือ สัญญาณเลือกการทำงานระหว่างเป็นพอร์ตปกติหรือเป็นขาเชื่อมต่ออุปกรณ์พิเศษ (PORT/PERIPHERAL Select) และสัญญาณควบคุมการส่งข้อมูลของวงจรถื่อมต่ออุปกรณ์ (Peripheral Output Enable) สำหรับข้อมูลของวงจรถื่อมต่ออุปกรณ์ที่ส่งออก

และรับเข้ามาจะผ่านทางขาพอร์ตปกติ แต่เมื่อผ่านวงจรสำหรับเลือกสัญญาณข้อมูลแล้ว สายสัญญาณข้อมูลของพอร์ต (data bus) กับข้อมูลของวงจรเชื่อมต่ออุปกรณ์ (peripheral output/peripheral input) จะแยกกัน

ไดอะแกรมของขาพอร์ต RC3-RC4 ทั้ง 2 ขานี้มีความพิเศษตรงที่สามารถใช้งานเป็นขาเชื่อมต่ออุปกรณ์อนุกรมแบบซิงโครนัส ซึ่งแบ่งเป็นระบบ SPI และระบบบัส I²C จึงทำให้ต้องเพิ่มสายสัญญาณควบคุมอินพุตเพิ่มเข้ามาอีก 1 เส้น เพื่อเลือกสัญญาณอินพุตระหว่าง SPI และ บัส

วงจรอินพุตบัฟเฟอร์ของขาพอร์ต C นี้เป็นแบบซิมิลาร์ทริกเกอร์ทั้งหมด ทั้งนี้เพื่อจัดการให้สัญญาณอินพุตที่เข้ามามีความเหมาะสมและสมบูรณ์มากที่สุด และยังคงสามารถรองรับการพูลอัปภายในแบบอัตโนมัติได้

3.7.4 พอร์ต D

มี 8 บิต กำหนดชื่อขาเป็น RD0-RD7 รีจิสเตอร์ที่ใช้ในการเก็บข้อมูลคือ PORTD มีแอดเดรสอยู่ที่ 0x08 (แบงก์ 0) เป็นรีจิสเตอร์ขนาด 8 บิต ส่วนการกำหนดทิศทางของพอร์ตนี้กระทำผ่านรีจิสเตอร์ TRISD มีแอดเดรสอยู่ที่ 0x88 (แบงก์ 1) มีขนาด 8 บิต หากต้องการกำหนดให้ขาพอร์ตในบิตใดเป็นอินพุตต้องเขียนข้อมูล "1" ไปยังบิตนั้น และในทางตรงข้ามหากต้องการกำหนดให้เป็นขาเอาต์พุต ให้เขียนข้อมูล "0" ไปยังบิตนั้น สำหรับพอร์ต D นี้จะมีเฉพาะในไมโครคอนโทรลเลอร์ PIC รุ่น 40 ขาเท่านั้น สำหรับในอนุกรม PIC16F87x จะมีในเบอร์ PIC16F871, PIC16F874(A) และ PIC16F877(A)

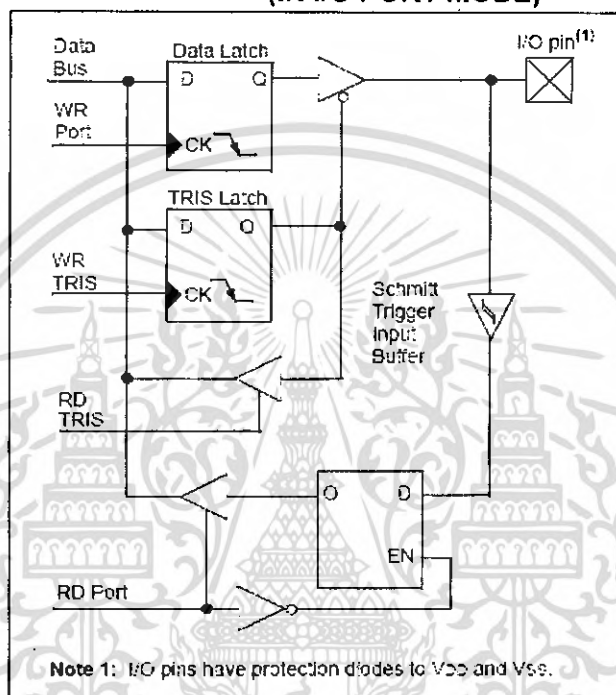
โครงสร้างทางฮาร์ดแวร์

พอร์ต D สามารถใช้งานได้ 2 โหมดคือ เป็นขาพอร์ตอินพุตเอาต์พุตปกติและเป็นส่วนขยายพอร์ตแบบขนาน (Parallel Slave Port : PSP) สำหรับเชื่อมต่อกับอุปกรณ์ภายนอกที่มีการจัดระบบบัสแบบไมโครโปรเซสเซอร์คือ มีสายข้อมูล 8 เส้น สายสัญญาณควบคุม 3 เส้นคือ สายสัญญาณควบคุมการอ่าน (RD : Read), เขียน (WR : Write) และเลือกอุปกรณ์ (CS : Chip Select)

ในรูปที่ 3.11 แสดงไดอะแกรมของพอร์ต D ซึ่งมีโครงสร้างเหมือนกันทุกบิต เมื่อทำงานในโหมดพอร์ตอินพุตเอาต์พุตปกติ วงจรอินพุตจะเป็นแบบซิมิลาร์ทริกเกอร์ แต่เมื่อทำงานในโหมดขยายพอร์ตแบบขนานหรือ PSP วงจรอินพุตจะเปลี่ยนเป็นแบบทีทีแอล

การเลือกโหมดทำงานของพอร์ต D นี้ขึ้นกับบิต PSPMODE (บิต 4 ในรีจิสเตอร์ TRISE) ถ้าเป็น “0” เป็นการกำหนดให้พอร์ต D เป็นพอร์ตปกติ และถ้าเป็น “1” พอร์ต D จะทำงานในโหมด PSP

**PORTD BLOCK DIAGRAM
(IN I/O PORT MODE)**



โครงสร้างของพอร์ต D ในไมโครคอนโทรลเลอร์ PIC16F87X

รูปที่ 3.11 โครงสร้างของขาพอร์ต D ในไมโครคอนโทรลเลอร์ PIC16F87x

3.7.5 พอร์ต E

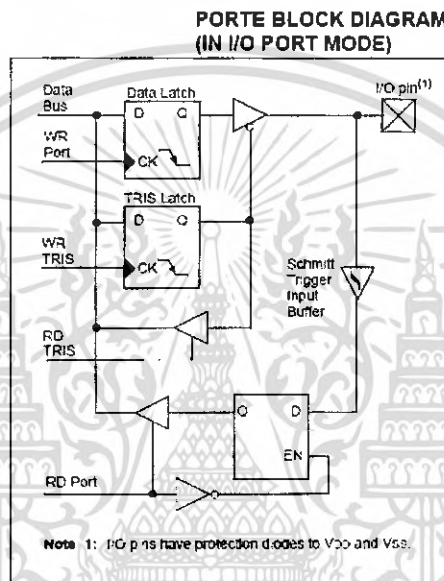
มี 3 บิต กำหนดชื่อขาเป็น RE0-RE2 รีจิสเตอร์ที่ใช้ในการเก็บข้อมูลคือ PORTE มีแอดเดรสอยู่ที่ 0x09 (แบงก์ 0) เป็นรีจิสเตอร์ขนาด 8 บิต แต่ใช้งานเพียง 3 บิตล่างคือ บิต 2 เท่านั้น ที่เหลือกำหนดให้เป็น “0” ส่วนการกำหนดทิศทางของพอร์ตนี้กระทำผ่านรีจิสเตอร์ TRISE ซึ่งมีแอดเดรสอยู่ที่ 0x89 (แบงก์ 1) มีขนาด 8 บิต โดยใช้สามบิตล่างในการกำหนดทิศทางของพอร์ต E ส่วนที่เหลือใช้ควบคุมการทำงานในโหมด PSP ของพอร์ต D

พอร์ต E สามารถใช้งานเป็นพอร์ตอินพุตเอาต์พุตปกติ, ขาอินพุตอะนาลอกของโมดูลแปลงสัญญาณอะนาลอกเป็นดิจิทัล และควบคุมการติดต่อกับอุปกรณ์ภายนอกแบบ PSP ทั้งนี้ขึ้นอยู่กับข้อกำหนดข้อมูลของรีจิสเตอร์ที่ใช้ควบคุมการทำงานของพอร์ตนี้ เช่นเดียวกับพอร์ต D

พอร์ต E จะมีเฉพาะในไมโครคอนโทรลเลอร์ PIC รุ่น 40 ขาเท่านั้น สำหรับในอนุกรม PIC16F87x จะมีในเบอร์ PIC16F871, PIC16F874(A) และ PIC16F877(A)

รีจิสเตอร์ TRIASE

เมื่อมีการกำหนดให้พอร์ต D ทำงานในโหมด PSP 4 บิต บนของรีจิสเตอร์ TRIASE จะใช้เป็นบิตแสดงสถานะของวงจรับัฟเฟอร์ที่ใช้ในการถ่ายทอดข้อมูลของส่วนขยายพอร์ตแบบขนาน หรือ PSP ส่วน 3 บิตล่างใช้ในการเลือกโหมดการทำงานของพอร์ต D



รูปที่ 3.12 โครงสร้างของขาพอร์ต E ในไมโครคอนโทรลเลอร์ PIC16F87x

ถ้าหากพอร์ต E ทำงานเป็นพอร์ตอินพุตเอาต์พุตปกติ 3 บิตล่างของรีจิสเตอร์ TRIASE จะใช้ในการกำหนดทิศทางการถ่ายทอดสัญญาณข้อมูลของพอร์ต ดังมีรายละเอียดการทำงานของแต่ละบิตของรีจิสเตอร์ TRIASE ดังนี้

IBF (Input Buffer Full status bit) : บิตแสดงสถานะบัฟเฟอร์ทางอินพุตของ PSP

“0” แสดงว่า ไม่มีข้อมูลรับเข้ามาในบัฟเฟอร์

“1” แสดงว่า มีข้อมูลเข้ามาและกำลังรอการอ่านจากซีพียู

OBF (Output Buffer Full status bit) : บิตแสดงสถานะบัฟเฟอร์ทางเอาต์พุตของ PSP

“0” แสดงว่า ข้อมูลในบัฟเฟอร์ถูกส่งออกไปแล้ว

“1” แสดงว่า ยังคงมีข้อมูลเดิมอยู่ในบัฟเฟอร์ทางเอาต์พุต

IBOV (Input Buffer Overflow Detect bit) : บิตแสดงสถานะบัฟเฟอร์อินพุต

“0” แสดงว่า ไม่มีการรับข้อมูลเกินหรือโอเวอร์โฟลวเกิดขึ้น

“1” แสดงว่า เกิดการเขียนข้อมูลมายังบัฟเฟอร์อินพุต ทั้งที่ยังคงมีข้อมูลเดิมอยู่

PSPMODE (Parallel Slave Port Mode select bit) บิตเลือกการทำงานของ PSP ของพอร์ต D และ E

“0” กำหนดให้ทำงานในโหมดพอร์ตอินพุตเอาต์พุตปกติ

“1” กำหนดให้ทำงานในโหมด PSP

บิต 3 : ไม่มีการใช้งาน กำหนดค่าเป็น “0”

DIR-RE2 ถึง DIR-RE0 (Direction control bit for RE2-RE0) บิตควบคุมทิศทางของพอร์ต E เมื่อทำงานในโหมดพอร์ตอินพุตเอาต์พุตปกติ

“0” เป็นเอาต์พุต

“1” เป็นอินพุต

โครงสร้างทางฮาร์ดแวร์

โคแอสเทรียมของพอร์ต E ในบิต RE0-RE2 เมื่อทำงานในโหมดพอร์ตอินพุตเอาต์พุตปกติ จะเห็นได้ว่ามีความคล้ายคลึงกับขาพอร์ตอื่นๆ ของ PIC16F87x ถ้าหากทำงานในโหมดพอร์ตอินพุตเอาต์พุตปกติ วงจรอินพุตของพอร์ตนี้จะเป็นแบบชนิดทรานซิสเตอร์ ในขณะที่หากทำงานในโหมดการแปลงสัญญาณอนาล็อกเป็นดิจิตอล วงจรอินพุตจะเปลี่ยนเป็นแบบทีทีแอล

ถึงแม้ว่าพอร์ต E ใน PIC16F87x มีจำนวนน้อยเพียง 3 บิต แต่สามารถเลือกรูปแบบการทำงานได้มากถึง 3 แบบ คือเป็นพอร์ตอินพุตเอาต์พุตปกติ, อินพุตสำหรับวงจรแปลงสัญญาณอนาล็อกเป็นดิจิตอลขนาด 10 บิตในไมโครคอนโทรลเลอร์ และพอร์ตสัญญาณควบคุมสำหรับติดต่อกับอุปกรณ์ในโหมด PSP ดังนั้นในการเลือกรูปแบบการทำงานต้องระมัดระวังเช่นเดียวกับพอร์ต C ที่ได้กล่าวมาแล้วก่อนหน้านี้

ในตารางที่ 3.2 แสดงการทำงานอย่างละเอียดของขาพอร์ต E ในโหมดต่างๆ

ชื่อขา	การทำงาน
RE0/RD/AN5	<ul style="list-style-type: none"> -ขาพอร์ตอินพุตเอาต์พุตบิต 0 ของพอร์ต E -ขาควบคุมการอ่านข้อมูลสำหรับส่วนขยายพอร์ตแบบขนานหรือโหมด PSP(RD) <ul style="list-style-type: none"> “0” = มีการอ่านข้อมูลเกิดขึ้น โดยทำการอ่านข้อมูลจากรีจิสเตอร์ PORTD ในกรณีที่มีการเลือกอุปกรณ์ที่ต้องการติดต่อด้วย (ขา CS = “0”) “1” = ไม่มีการอ่านข้อมูลเกิดขึ้น -อินพุตรับสัญญาณอะนาลอกช่อง 5
RE1/WR/AN6	<ul style="list-style-type: none"> -ขาพอร์ตอินพุตเอาต์พุตบิต 1 ของพอร์ต E -ขาควบคุมการเขียนข้อมูลสำหรับส่วนขยายพอร์ตแบบขนานหรือโหมด PSP(RD) <ul style="list-style-type: none"> “0” = มีการเขียนข้อมูลเกิดขึ้น โดยเขียนข้อมูลไปยังจากรีจิสเตอร์ PORTD ในกรณีที่มีการเลือกอุปกรณ์ที่ต้องการติดต่อด้วย (ขา CS = “0”) “1” = ไม่มีการเขียนข้อมูลเกิดขึ้น -อินพุตรับสัญญาณอะนาลอกช่อง 6
RE2/CS/AN7	<ul style="list-style-type: none"> - ขาพอร์ตอินพุตเอาต์พุตบิต 2 ของพอร์ต E - ขาเลือกอุปกรณ์ที่ต้องการติดต่อด้วยสำหรับส่วนขยายพอร์ตแบบขนานหรือ PSP (CS) <ul style="list-style-type: none"> “0” = มีการเลือกอุปกรณ์ “1” = ไม่มีการเลือกอุปกรณ์ - อินพุตรับสัญญาณอะนาลอกช่อง 7

ตารางที่ 3.2 แสดงหน้าที่การทำงานของขาพอร์ต E ในไมโครคอนโทรลเลอร์ PIC16F87x

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

ทฤษฎีของสเต็ปปีงมอเตอร์ (Stepping Motor)

4.1 บทนำ

โดยทั่วไปแล้วมอเตอร์ไฟฟ้ากระแสตรง (DC Motor) หรือมอเตอร์ไฟฟ้ากระแสสลับ (AC Motor) จะมีการหมุนแบบต่อเนื่อง ไม่สามารถควบคุมการหมุนที่เป็นองศาหรือระยะทางได้ ดังนั้น สเต็ปปีงมอเตอร์ (Stepping Motor) จึงเป็นทางเลือกอีกทางหนึ่งที่สามารถควบคุมทิศทางการหมุนที่เป็นองศาหรือระยะทางตามความต้องการได้ ลักษณะการทำงานของสเต็ปปีงมอเตอร์ จะต้องทำการป้อนสัญญาณพัลส์ (Pulse) ให้กับขด สเตเตอร์ (Stator) ทำให้เกิดแรงผลักร็โรเตอร์ (Rotor) จึงเกิดการหมุน โดยปกติแล้วการหมุนของสเต็ปปีงมอเตอร์ จะหมุนเป็นจังหวะ (Step) ตามสัญญาณพัลส์ที่ป้อนเข้ามา

การหมุนของสเต็ปปีงมอเตอร์ เมื่อหมุนครบ 1 รอบ จะเท่ากับ 360 องศา ดังนั้น ถ้าสเต็ปปีงมอเตอร์ ตัวหนึ่งมีการหมุนเท่ากับ 5 องศาต่อสเต็ป ความละเอียดของการหมุนของสเต็ปปีงมอเตอร์ตัวนี้ จะเท่ากับ 72 สเต็ปต่อ 1 รอบ โดยทั่วไปแล้ว สเต็ปปีงมอเตอร์จะถูกนำไปใช้งานอย่างแพร่หลายในปัจจุบัน เพราะสามารถควบคุมตำแหน่งของการหมุนตำแหน่งใดก็ได้ เช่น หัวอ่าน CD ROM, Tape Classes เป็นต้นตลอดจนในอุตสาหกรรมการผลิตต่างๆ เช่น แขนกลในงานอุตสาหกรรม, ระบบสายพาน เป็นต้น

ในกรณีที่จ่ายกระแสไฟฟ้าให้กับขดลวดขดนั้นของสเต็ปปีงมอเตอร์ตลอดเวลา จะเกิดการหมุนเพียงหนึ่งสเต็ปเท่านั้น ซึ่งจะแตกต่างจากมอเตอร์ไฟฟ้ากระแสตรงที่เมื่อจ่ายกระแสไฟฟ้าให้ก็จะทำให้หมุนจนกว่าจะทำการหยุดจ่ายกระแสไฟฟ้า

ข้อดีของสเต็ปปีงมอเตอร์ เมื่อเปรียบเทียบกับมอเตอร์ไฟฟ้ากระแสตรง

1. สามารถใช้ในงานควบคุมตำแหน่งในลักษณะ วงจรควบคุมแบบเปิด (Open Loop Control) ได้ โดยที่ไม่ต้องการสัญญาณป้อนกลับ (Feed Back Signal) แต่จะอาศัยการนับจำนวนของพัลส์ที่ส่งไปควบคุมการหมุนแทน

2. ไม่มีส่วนของแปร่งถ่านที่จะทำให้เกิดการสึกหรอ และจะไม่เกิดการสปาร์กที่แปร่งถ่าน ซึ่งอาจทำให้เกิดสัญญาณรบกวนได้

4.2 ชนิดของสเต็ปป์มอเตอร์

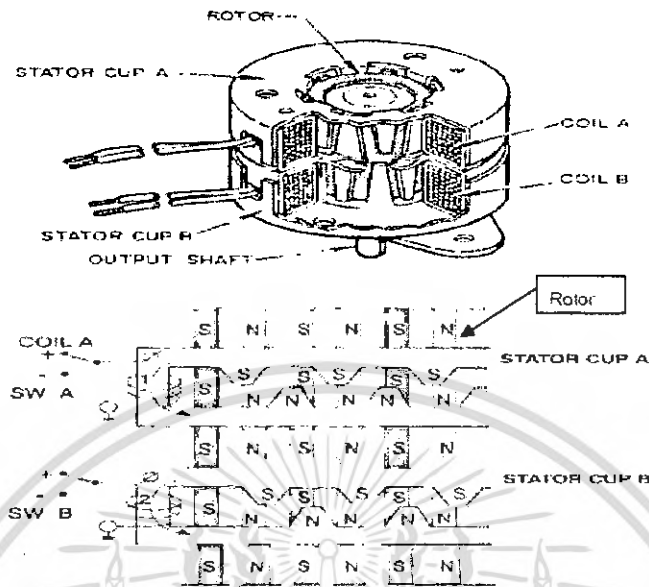
สเต็ปป์มอเตอร์ที่ใช้งานอยู่ทั่วไปสามารถแบ่งออกได้เป็น 3 แบบ คือ

- แบบแม่เหล็กถาวร (Permanent Magnet : PM)
- แบบเปลี่ยนแปลงค่ารีลักแทนซ์ (Variable Reluctance : VR)
- แบบผสม (Hybrid : HB)

4.2.1 แบบแม่เหล็กถาวร (Permanent Magnet : PM)

โครงสร้างภายในของสเต็ปป์มอเตอร์แบบแม่เหล็กถาวรแสดงดังรูปที่ 1 ซึ่งส่วนของโรเตอร์ จะทำด้วยแม่เหล็กถาวรรูปทรงกระบอกเรียบ ในส่วนของสเตเตอร์จะมีขดลวดพันอยู่ และทำเป็นลักษณะของฟัน (Teeth) เมื่อจ่ายกระแสไฟฟ้าให้กับขดลวดที่สเตเตอร์จะเกิดเป็นขั้วแม่เหล็กขึ้นที่ฟันของสเตเตอร์ และจะดึงดูดกับขั้วแม่เหล็กถาวรที่โรเตอร์ ทำให้เกิดการหมุนของโรเตอร์ขึ้น โดยทั่วไปแล้วสเต็ปป์มอเตอร์แบบแม่เหล็กถาวรนี้จะมีตั้งแต่ขนาด 2 เฟสขึ้นไป และไม่นิยมนำไปใช้งานอุตสาหกรรม แต่จะถูกนำไปใช้กับอุปกรณ์คอมพิวเตอร์ เช่น ตัวขับวงล้อที่ใช้หมุนเพื่อเลื่อนกระดาษของเครื่องพิมพ์, CD ROM เป็นต้น เพราะความเร็วในการทำงานต่ำ แรงบิดต่ำ และนอกจากนี้ด้วยโครงสร้างของสเต็ปป์มอเตอร์ชนิดนี้ทำให้มุมที่หมุนไปแต่ละสเต็ปไม่ละเอียด เช่น สเต็ปละ 3.6 , 7.5 , 15 , 18 องศา เป็นต้น

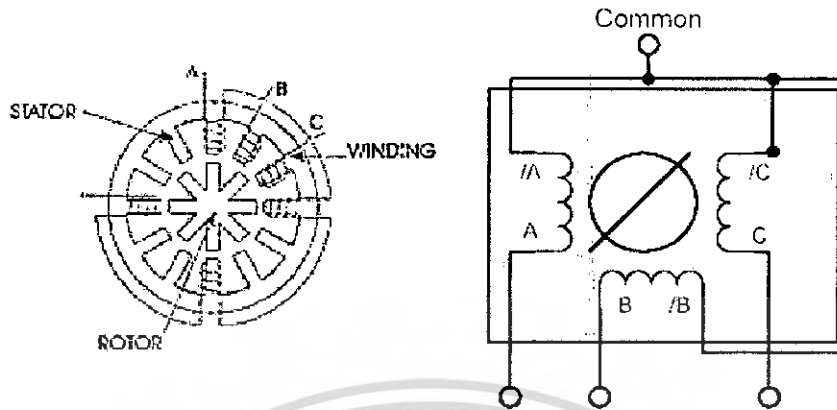
ในขณะที่ไม่มีการจ่ายกระแสไฟฟ้าให้กับขดลวดบนสเตเตอร์ของสเต็ปป์มอเตอร์ชนิดนี้ ตัวโรเตอร์ก็จะเกิดแรงดึงดูดกับสเตเตอร์ ซึ่งเกิดจากอำนาจของแม่เหล็กถาวรที่โรเตอร์ ทำให้หมุนได้ยาก ดังนั้น เราสามารถสังเกต สเต็ปป์มอเตอร์ชนิดนี้ได้ โดยการทดลองหมุนดูด้วยมือเปล่าจะรู้สึกว่ามีแรงเสียดทานเป็นขั้นๆ เพราะการเหนี่ยวนำของแม่เหล็กถาวรที่โรเตอร์



รูปที่ 4.1 โครงสร้างของสเต็ปปีงมอเตอร์แบบแม่เหล็กถาวร

4.2.2 แบบเปลี่ยนแปลงค่ารีลักแทนซ์ (Variable Reluctance : VR)

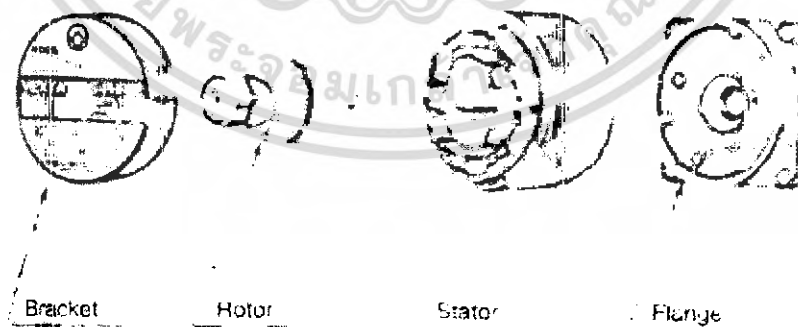
โครงสร้างของสเต็ปปีงมอเตอร์ค่ารีลักแทนซ์แสดงดังรูปที่ 2 ส่วนของโรเตอร์จะทำด้วยเหล็กอ่อนรูปทรงกระบอก และทำเป็นลักษณะของฟัน ในส่วนของสเตเตอร์จะมีขดลวดพันอยู่และจะทำเป็นลักษณะของฟันเช่นกัน เมื่อทำการจ่ายกระแสให้ขดลวดที่สเตเตอร์จะเกิดเป็นขั้วแม่เหล็กที่ฟันสเตเตอร์และเหนี่ยวนำกับโรเตอร์ ทำให้ฟันของโรเตอร์เกิดเป็นขั้วแม่เหล็กที่มีขั้วตรงกันข้ามกับสเตเตอร์ จึงทำให้มีแรงดึงดูดซึ่งกันและกัน เกิดการหมุนของโรเตอร์ขึ้น สเต็ปปีงมอเตอร์ชนิดนี้โดยทั่วไปแล้วจะมีขนาด 3 เฟส แต่ในบางครั้งอาจพบขนาด 4 เฟส และไม่นิยมนำไปใช้ในงานอุตสาหกรรม แต่จะถูกนำไปใช้ในงานที่มีขนาดเล็กเช่นเดียวกับสเต็ปปีงมอเตอร์แบบแม่เหล็กถาวร เช่น Micro-positioning table เป็นต้น เพราะไม่มีส่วนที่เป็นแม่เหล็กถาวร ดังนั้นในขณะที่ไม่มีจ่ายกระแสไฟฟ้าให้กับขดลวดที่สเตเตอร์จึงไม่เกิดแรงดึงดูดระหว่างโรเตอร์กับสเตเตอร์



รูปที่ 4.2 โครงสร้างของสเต็ปป์มอเตอร์แบบเปลี่ยนแปลงค่ารีลักแทนซ์และสัญลักษณ์

4.2.3 แบบผสม (Hybrid : HB)

Stepping Motor ชนิดนี้จะใช้หลักการทำงานของสเต็ปป์มอเตอร์แบบเปลี่ยนแปลงค่ารีลักแทนซ์ และ สเต็ปป์มอเตอร์แบบแม่เหล็กถาวร มาออกแบบโดยที่สเตเตอร์จะคล้ายกับ Stepping Motor แบบเปลี่ยนแปลงค่ารีลักแทนซ์ ส่วนโรเตอร์จะคล้ายกับสเต็ปป์มอเตอร์แบบแม่เหล็กถาวร แต่จะทำเป็นลักษณะของฟัน สเต็ปป์มอเตอร์ชนิดนี้โดยทั่วไปจะนิยมใช้ในงานอุตสาหกรรม เพราะมีแรงบิดสูง ความละเอียดของสเต็ปในการหมุนสูง มีความเร็วในการทำงานสูงกว่าสเต็ปป์มอเตอร์สองแบบที่กล่าวมา และสเต็ปป์มอเตอร์ชนิดนี้โดยทั่วไปจะมีขนาด 2 เฟส ถึง 5 เฟส



รูปที่ 4.3 แสดงสเต็ปป์มอเตอร์แบบผสมขนาด 5 เฟส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

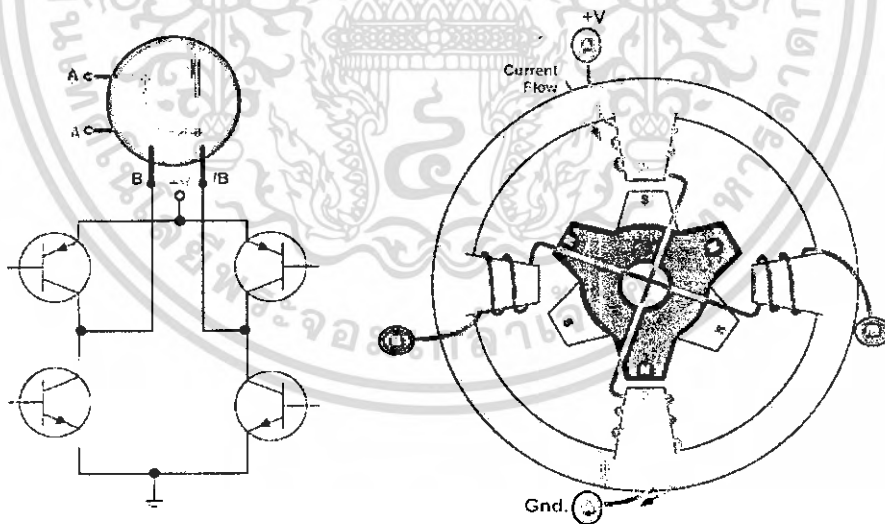
4.3 การพันขดลวดของสเต็ปิ่งมอเตอร์

สเต็ปิ่งมอเตอร์โดยทั่วไปจะมีการพันขดลวดอยู่ 2 แบบ คือ

- แบบไบโพลาร์ (Bipolar)
- แบบยูนิโพลาร์ (Unipolar)

4.3.1 แบบไบโพลาร์ (Bipolar)

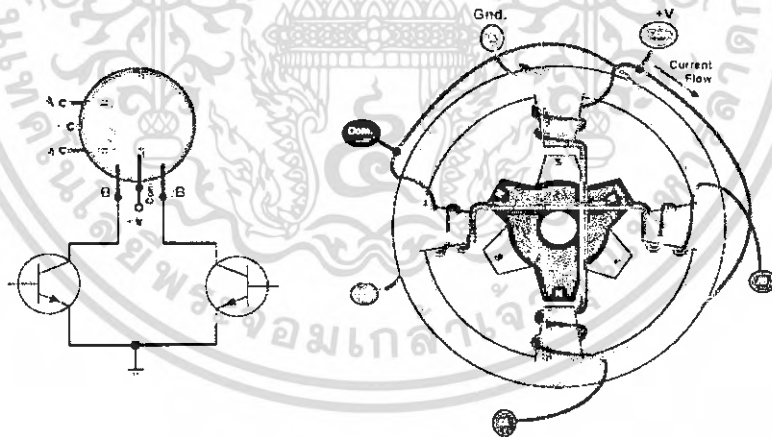
การพันขดลวดของสเต็ปิ่งมอเตอร์ ในลักษณะนี้จะมีการพันขดลวด 1 ขด บนขั้วแม่เหล็กของขดลวดสเตเตอร์ ขดลวดที่สเตเตอร์แต่ละขดจะไม่มีจุกพร้อม การต่อเข้ากับวงจรขับจะใช้ปลายทั้งสองด้านของขดลวดแต่ละขด การทำให้เกิดขั้วแม่เหล็กที่สเตเตอร์ ทำได้โดยการจ่ายกระแสไฟฟ้าจากปลายด้านหนึ่งไปยังปลายอีกด้านหนึ่งของขดลวด ขั้วแม่เหล็กของสเตเตอร์จะถูกกำหนดโดยทิศทางการไหลของกระแสไฟฟ้า ซึ่งการทำให้ขั้วแม่เหล็กเกิดทิศทางการกันข้ามทำได้โดยการกลับทิศทางการไหลของกระแสไฟฟ้า ดังนั้นวงจรขับที่ใช้จึงจำเป็นต้องสามารถกลับทิศทางการไหลของกระแสไฟฟ้าได้ โดยทั่วไปแล้วการพันขดลวดแบบ Bipolar จะทำให้เกิดแรงบิดมากกว่าแบบ Unipolar



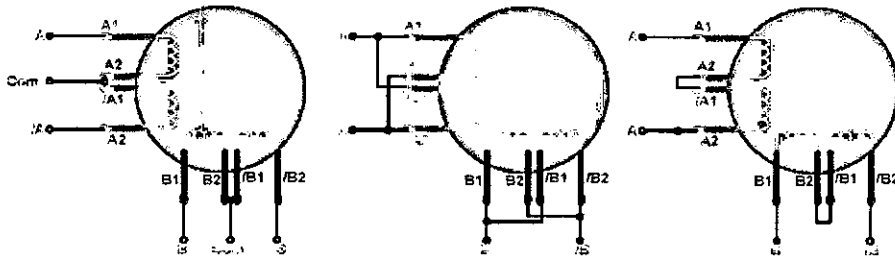
รูปที่ 4.4 แสดงการพันขดลวดและวงจรสวิตชิงแบบ Bipolar

4.3.2 แบบยูนิโพลาร์ (Unipolar)

การพันขดลวดของสเต็ปิ้งมอเตอร์ ในลักษณะนี้จะมีการพันขดลวด 2 ขด บนแต่ละขั้วแม่เหล็กของขดลวดสเตเตอร์ ขดลวดที่สเตเตอร์แต่ละขดจะมีจุดร่วม การต่อเข้ากับวงจรขับจะใช้ปลายของขดลวดแต่ละด้านต่อเข้ากับวงจรขับและใช้จุดร่วมต่อเข้ากับขั้วบวกของแหล่งจ่ายไฟ การทำให้เกิดขั้วแม่เหล็กที่สเตเตอร์ทำได้โดยการจ่ายกระแสไฟฟ้าให้ไหลจากจุดร่วมลงกราวด์มาครบวงจรที่ปลายด้านหนึ่งของขดลวด การเปลี่ยนขั้วแม่เหล็กที่สเตเตอร์ชุดเดียวกันนี้ ก็ทำได้โดยเปลี่ยนการจ่ายกระแสไฟฟ้าจากขดหนึ่งไปยังอีกขดหนึ่งของขดลวดที่พันอยู่บนสเตเตอร์ชุดเดียวกัน ดังนั้นวงจรขับจึงต้องเป็นวงจรสวิตช์ เพื่อที่จะสามารถกลับทิศทางการไหลของกระแสไฟฟ้าได้และเพื่อทำให้การจ่ายกระแสไฟฟ้าที่ไหลผ่านขดลวดนั้นครบวงจร ขดลวดแต่ละขดจะแบ่งออกเป็น 2 เฟส รวมทั้งหมดมี 4 เฟส คือ 1, 2, 3 และ 4 โดยทั่วไปแล้วการพันขดลวดของสเต็ปิ้งมอเตอร์แบบนี้ จะมีแบบ 5 สาย 4 เฟส และ 6 สาย 4 เฟส โดยแบบ 5 สาย 4 เฟส จะมีสายออกมาจากตัวสเต็ปิ้งมอเตอร์ 6 เส้น โดยสาย 4 เส้นแรกจะต่อที่เฟสของสเต็ปิ้งมอเตอร์ และอีก 2 เส้น (Common) จะนำมาต่อรวมกัน



รูปที่ 4.5 แสดงการพันขดลวดและวงจรสวิตช์ซึ่งแบบ Unipolar



(ก) Unipolar Motor

(ข) Bipolar Motor (Parallel)

(ค) Bipolar Motor (Series)

รูปที่ 4.6 แสดงสัญลักษณ์การต่อสเต็ปมอเตอร์แบบต่างๆ

4.4 การควบคุมการหมุนของสเต็ปมอเตอร์

การทำให้สเต็ปมอเตอร์หมุนได้นั้น จะต้องมีการป้อนสัญญาณพัลส์ ที่ขดลวดแต่ละเฟส ของสเต็ปมอเตอร์ให้ไปในทิศทางเดียวกันหรือเรียงลำดับ (Sequence) กัน และถ้าต้องการให้สเต็ป มอเตอร์หมุนกลับทิศทางก็จะต้องป้อนสัญญาณพัลส์ในทิศทางตรงกันข้าม

การควบคุมการหมุนของสเต็ปมอเตอร์สามารถแบ่งออกได้เป็น 3 แบบ คือ

- แบบฟลูตเต็ป 1 เฟส (Full – Step 1 Phase)
- แบบฟลูตเต็ป 2 เฟส (Full – Step 2 Phase)
- แบบฮาล์ฟสเต็ป (Half Step)

4.4.1 แบบฟลูตเต็ป 1 เฟส (Full – Step 1 Phase)

การควบคุมในลักษณะนี้จะเป็นการกระตุ้นขดลวดทีละขดเรียงตามลำดับ 1, 2, 3, 4 การ กระตุ้นจะมีขดลวดขดเดียวในเวลาหนึ่งที่ถูกกระตุ้นเท่านั้น เช่น ขดที่ 1, 2, 3, 4 หรือถ้าต้องการให้ หมุนกลับทิศทางก็ให้กระตุ้นขดลวดที่ 4, 3, 2, 1 เป็นต้น การกระตุ้นแบบนี้จะทำงานง่ายที่สุดและ กินกระแสไฟฟ้าน้อยที่สุดด้วย

ตารางที่ 4.1 แสดงการควบคุมสเต็ปปีงมอเตอร์แบบ Full – Step 1 Phase

สเต็ปที่	เฟสที่ 1	เฟสที่ 2	เฟสที่ 3	เฟสที่ 4
1	ทำงาน	-	-	-
2	-	ทำงาน	-	-
3	-	-	ทำงาน	-
4	-	-	-	ทำงาน

4.4.2 แบบฟูลสเต็ป 2 เฟส (Full – Step 2 Phase)

การควบคุมในลักษณะนี้จะเป็นการกระตุ้นขดลวดทีละ 2 ขด ที่อยู่ใกล้กันให้ทำงานพร้อมกัน และเรียงลำดับกันไป เช่น ขดที่ 12 , 23 , 34 , 41 หรือถ้าต้องการให้มีการหมุนกลับทิศทางก็ให้ทำการกระตุ้นขดลวดที่ 14 , 43 , 32 , 21 เป็นต้น ข้อดีของการกระตุ้นแบบนี้ คือ สเต็ปปีงมอเตอร์จะมีแรงบิดมากกว่าแบบ 1 เฟส ส่วนข้อเสียของการกระตุ้นแบบนี้ คือ จะต้องใช้กำลังไฟฟ้าเพิ่มขึ้นเป็น 2 เท่าของแบบ 1 เฟส เพราะจะต้องกระตุ้นขดลวด 2 ขดลวดพร้อมกัน

ตารางที่ 4.2 แสดงการควบคุมสเต็ปปีงมอเตอร์แบบ Full – Step 2 Phase

สเต็ปที่	เฟสที่ 1	เฟสที่ 2	เฟสที่ 3	เฟสที่ 4
1	ทำงาน	ทำงาน	-	-
2	-	ทำงาน	ทำงาน	-
3	-	-	ทำงาน	ทำงาน
4	ทำงาน	-	-	ทำงาน

4.4.3 แบบฮาล์ฟสเต็ป (Half Step)

การควบคุมในลักษณะนี้จะเป็นการกระตุ้นแบบ 1 เฟส กับ 2 เฟส มารวมกัน โดยการกระตุ้นจะเรียงลำดับกันไป เช่น ขดที่ 1 , 12 , 2 , 23 , 3 , 34 , 4 , 41 หรือถ้าต้องการให้มีการหมุนกลับทิศทางก็ให้ทำการกระตุ้นขดลวดที่ 1 , 41 , 4 , 43 , 3 , 32 , 2 , 21 เป็นต้น ข้อดีของการกระตุ้นแบบนี้ คือ สเต็ปปีงมอเตอร์จะมีแรงบิดเพิ่มมากขึ้น ละเอียดขึ้นและการควบคุมตำแหน่งถูกต้องมากยิ่งขึ้น

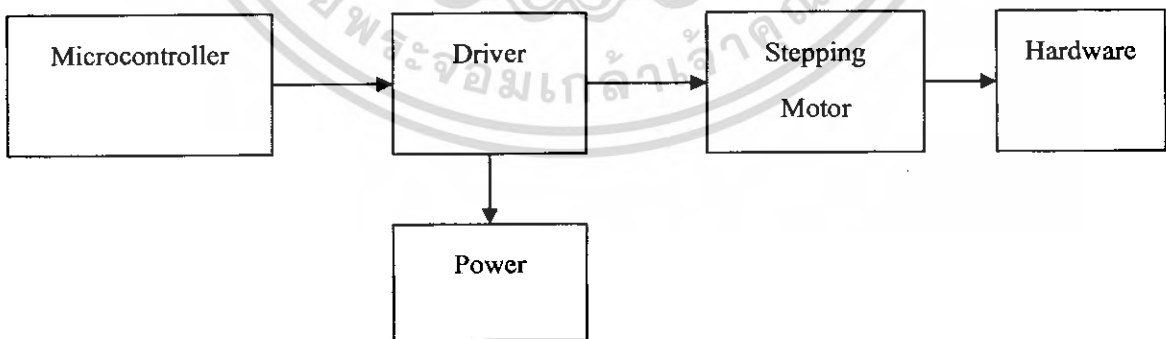
ส่วนข้อเสียของการกระตุ้นแบบนี้ คือ จะต้องใช้กำลังไฟฟ้าเพิ่มขึ้นเป็น 2 เท่า เหมือนกับการกระตุ้นแบบ 2 เฟสนั่นเอง

ตารางที่ 4.3 แสดงการควบคุมสเต็ปป์มอเตอร์แบบ Half Phase

สเต็ปที่	เฟสที่ 1	เฟสที่ 2	เฟสที่ 3	เฟสที่ 4
1	ทำงาน	-	-	-
2	ทำงาน	ทำงาน	-	-
3	-	ทำงาน	-	-
4	-	ทำงาน	ทำงาน	-
5	-	-	ทำงาน	-
6	-	-	ทำงาน	ทำงาน
7	-	-	-	ทำงาน
8	ทำงาน	-	-	ทำงาน

4.5 การควบคุมการทำงานของสเต็ปป์มอเตอร์

ในส่วนนี้จะเป็นการยกตัวอย่างการใช้ Microcontroller เป็นตัวควบคุมการทำงาน เพื่อให้ สเต็ปป์มอเตอร์ทำงานโดยผ่านชุดขับ (Driver) เพื่อเป็นตัวสร้างสัญญาณขับมอเตอร์ในแต่ละเฟส ให้ทำงานได้ตามต้องการ โดยสามารถให้มีการเคลื่อนที่เป็นองศาหรือระยะทาง การควบคุมสเต็ปป์มอเตอร์สามารถเขียนเป็นบล็อกไดอะแกรมได้ ดังนี้

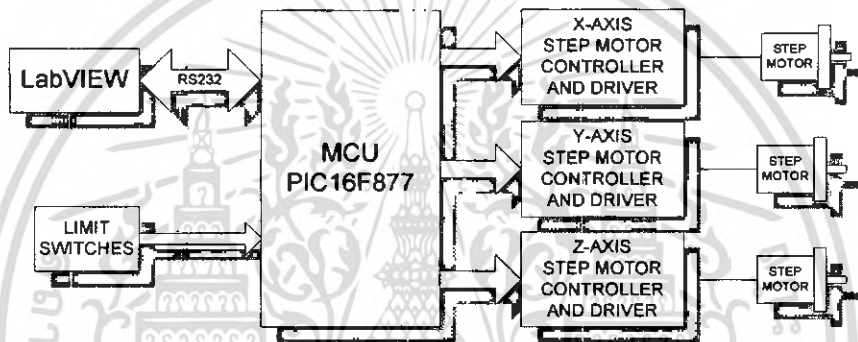


รูปที่ 4.7 แสดงไดอะแกรมการควบคุมสเต็ปป์มอเตอร์

บทที่ 5

การออกแบบและการทำงานของวงจร

ในการออกแบบวงจรจะมีส่วนสำคัญอยู่ด้วยกัน 2 ส่วนใหญ่ ๆ คือส่วนแรกจะเป็นส่วนของวงจรไมโครคอนโทรลเลอร์ (Microcontroller) ที่ใช้ในการติดต่อกับคอมพิวเตอร์และติดต่อกับวงจรควบคุม ส่วนที่สองจะเป็นส่วนที่ใช้ในการควบคุมรูปแบบการหมุนของสเต็ปมอเตอร์เพื่อสร้างสัญญาณไปที่หัวจรจับซึ่งมีโครงสร้างการทำงานดังรูปที่ 5.1



รูปที่ 5.1 โครงสร้างการทำงานในส่วนต่าง ๆ

ในส่วนของการติดต่อกับคอมพิวเตอร์จะใช้ไมโครคอนโทรลเลอร์ PIC16F877 ในการติดต่อผ่านทาง RS-232 โดยมีไอซี MAX232 ในการปรับระดับสัญญาณ โดยจะทำการส่งงานผ่านทางโปรแกรม LabVIEW

ในส่วนของการควบคุมการหมุนของสเต็ปมอเตอร์จะใช้ไอซีเบอร์ L297 และในส่วนของการจับสเต็ปมอเตอร์ จะใช้ไอซีเบอร์ L298 ซึ่งในการใช้งานจะใช้ไอซีทั้งสองเบอร์นี้คู่กัน ซึ่งการทำงานของไอซีทั้งสองเบอร์จะทำงานสัมพันธ์กัน และมีรายละเอียดต่าง ๆ ดังนี้

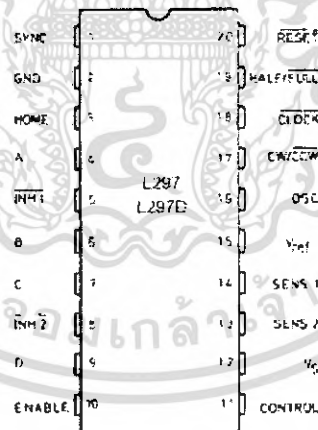
5.1 รายละเอียดของไอซี L297

- สามารถขับได้ทั้งแบบ ฟูลสเต็ป (FULL STEP) และ แบบฮาล์ฟสเต็ป (HALF STEP)
- สามารถควบคุมทิศทางการหมุนได้
- สามารถควบคุมกระแสเอาต์พุตได้
- สามารถเลือกได้ว่าจะให้ไอซีทำงานหรือให้หยุดทำงาน

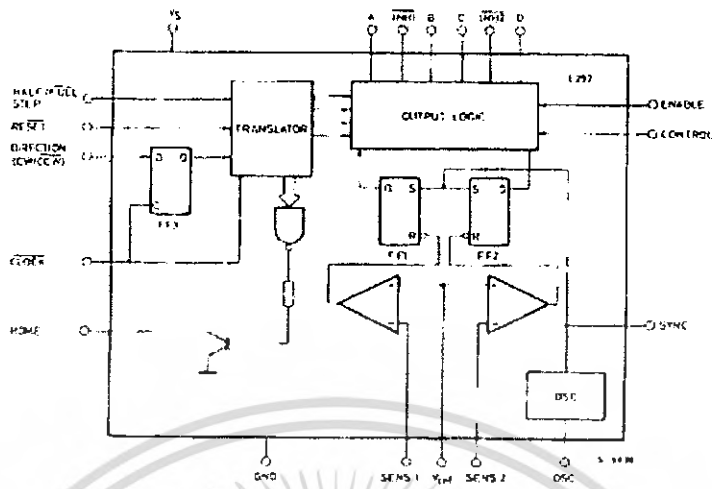
สามารถทำการควบคุมสเต็ปมอเตอร์ได้ทั้งแบบ ยูนิโพลาร์ (Unipolar) และ แบบไบโพลาร์ (Bipolar) สามารถควบคุมการหมุนได้ทั้งแบบ ฟูลสเต็ป และ ฮาล์ฟสเต็ป ควบคุมทิศทางการหมุนได้โดยการควบคุมด้วยการป้อนลอจิก ในการควบคุมการหมุนไปในแต่ละสเต็ปทำได้โดยการป้อนสัญญาณคล็อก (CLOCK) โดยในแต่ละสเต็ปที่ได้ออกจากเอาต์พุตของ L297 จะทำงานในช่วงลอจิกต่ำของสัญญาณคล็อกเพื่อที่จะใช้ในการควบคุมสัญญาณเอาต์พุต และ เพื่อที่จะนำสัญญาณที่ได้ไปเข้าส่วนของวงจรขับ การควบคุมกระแสเอาต์พุตทำได้โดยการป้อนแรงดันอ้างอิง โดยกระแสทางเอาต์พุตจะเป็นไปตามสมการ

$$I_o = V_{ref} / 0.5\Omega \text{ (Ohm)}$$

ในการควบคุมการเคลื่อนที่ของสเต็ป ความเร็วในการเคลื่อนที่และจำนวนของสเต็ปในการเคลื่อนที่ จะสัมพันธ์กับจำนวนและความถี่ของคล็อกที่ป้อน

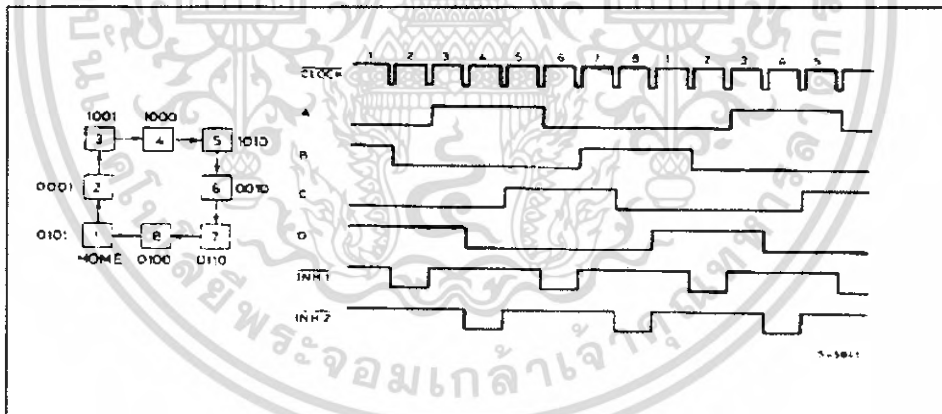


รูปที่ 5.2 ตำแหน่งขาของไอซี L297



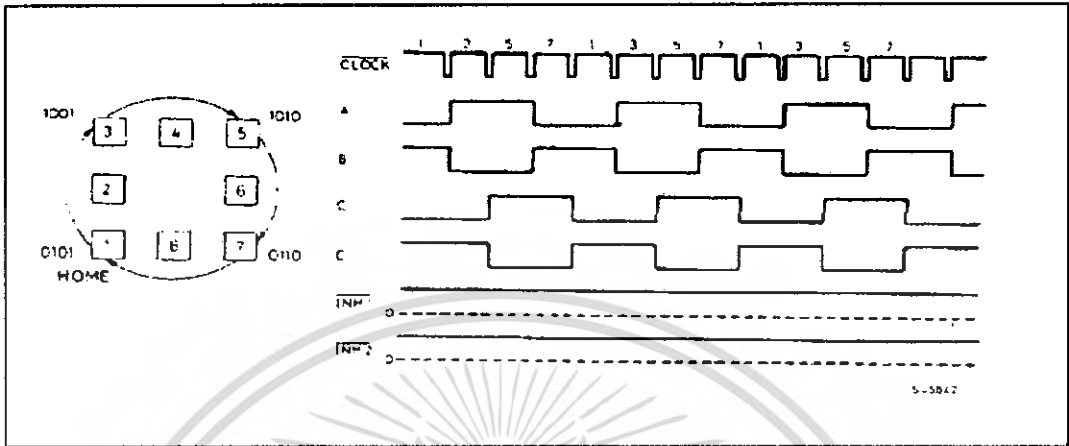
รูปที่ 5.3 โครงสร้างภายในไอซี L297

การขับในโหมด (Mode) ฮาล์ฟสเต็ป ทำได้โดยการป้อนลอจิกสูงเข้าที่ขา 19 โดยสัญญาณควบคุมต่าง ๆ จะเป็นดังรูปที่ 5.4

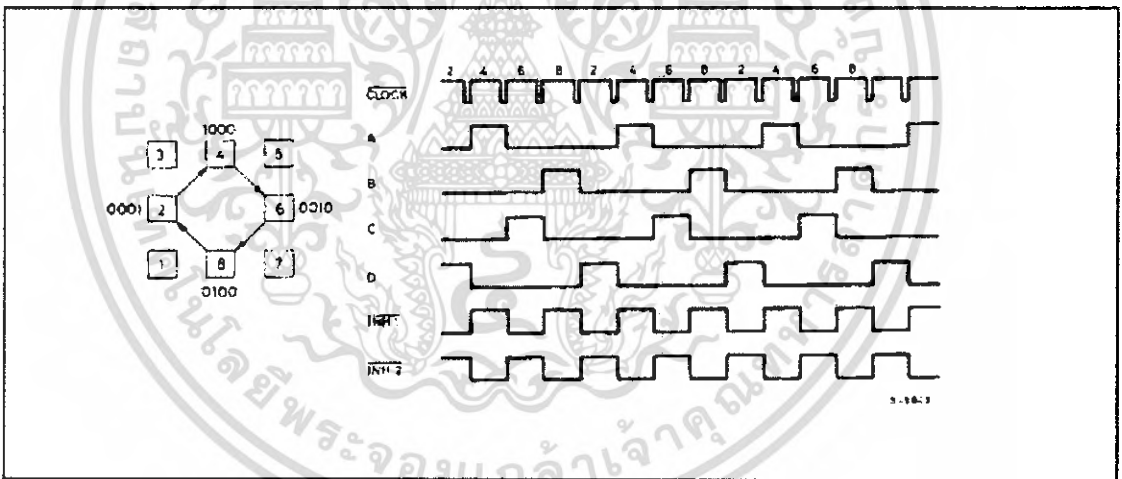


รูปที่ 5.4 สัญญาณที่ใช้ขับในโหมดฮาล์ฟสเต็ป

การขับในโหมด One-Phase และ Two-Phase ทำได้โดยการป้อนลอจิกค่าเข้าที่ขา 19 โดย
สัญญาณ ความคุมจะเป็นดังรูปที่ 5.5



รูปที่ 5.5 สัญญาณที่ใช้ขับในโหมด Two Phase



รูปที่ 5.6 สัญญาณที่ใช้ขับในโหมด One Phase

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รายละเอียดคุณสมบัติของขาต่าง ๆ

N°	NAME	FUNCTION
1	SYNC	Output of the on-chip chopper oscillator. The SYNC connections of all L297s to be synchronized are connected together and the oscillator components are omitted on all but one. If an external clock source is used it is injected at this terminal.
2	GND	Ground connection.
3	HOME	Open collector output that indicates when the L297 is in its initial state (ABCD = 0101). The transistor is open when this signal is active.
4	A	Motor phase A drive signal for power stage.
5	$\overline{\text{INH1}}$	Active low inhibit control for driver stage of A and B phases. When a bipolar bridge is used this signal can be used to ensure fast decay of load current when a winding is de-energized. Also used by chopper to regulate load current if CONTROL input is low.
6	B	Motor phase B drive signal for power stage.
7	C	Motor phase C drive signal for power stage.
8	$\overline{\text{INH2}}$	Active low inhibit control for drive stages of C and D phases. Same functions as INH1.
9	D	Motor phase D drive signal for power stage.
10	ENABLE	Chip enable input. When low (inactive) INH1, INH2, A, B, C and D are brought low.
11	CONTROL	Control input that defines action of chopper. When low chopper acts on INH1 and INH2; when high chopper acts on phase lines ABCD.
11	CONTROL	Control input that defines action of chopper. When low chopper acts on INH1 and INH2; when high chopper acts on phase lines ABCD.
12	V_s	5V supply input.
13	SENS ₂	Input for load current sense voltage from power stages of phases C and D.
14	SENS ₁	Input for load current sense voltage from power stages of phases A and B.
15	V_{ref}	Reference voltage for chopper circuit. A voltage applied to this pin determines the peak load current.
16	OSC	An RC network (R to V_{CC} , C to ground) connected to this terminal determines the chopper rate. This terminal is connected to ground on all but one device in synchronized multi-L297 configurations. $f = 1/0.59 RC$
17	$\overline{\text{CW/CCW}}$	Clockwise/counterclockwise direction control input. Physical direction of motor rotation also depends on connection of windings. Synchronized internally therefore direction can be changed at any time.
18	$\overline{\text{CLOCK}}$	Step clock. An active low pulse on this input advances the motor one increment. The step occurs on the rising edge of this signal.

รูปที่ 5.7 แสดงรายละเอียดของขาต่าง ๆ ของไอซี L297

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

N°	NAME	FUNCTION
19	HALF/FULL	Half/full step select input. When high selects half step operation, when low selects full step operation. One-phase-on full step mode is obtained by selecting FULL when the L297's translator is at an even-numbered state. Two-phase-on full step mode is set by selecting FULL when the translator is at an odd numbered position. (The home position is designate state 1).
20	RESET	Reset input. An active low pulse on this input restores the translator to the home position (state 1, ABCD = 0101).

รูปที่ 5.7 แสดงรายละเอียดของขาต่าง ๆ (ต่อ)

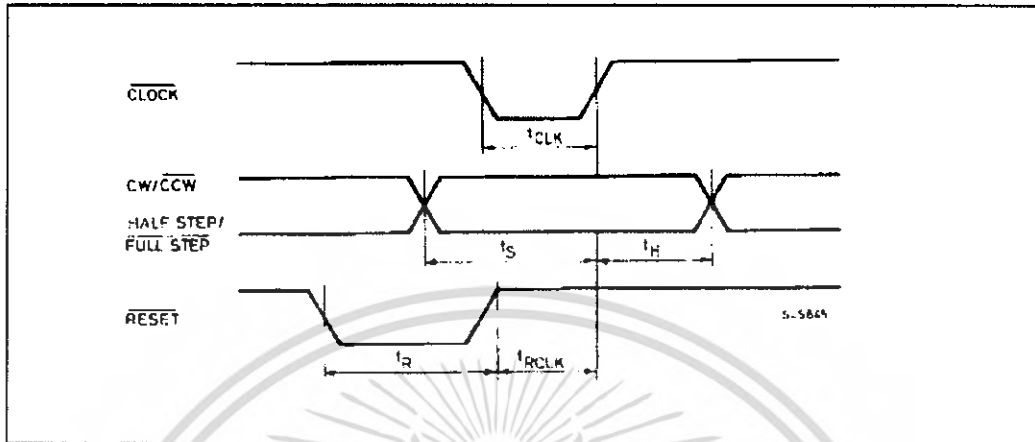
คุณสมบัติทางไฟฟ้าของ L297 ที่อุณหภูมิ 25°C แรงดันแหล่งจ่าย 5 V.

Symbol	Parameter	Test conditions	Min.	Typ	Max.	Unit
I_{pak}	Leakage current (pin 3)	$V_{\text{CE}} = 7 \text{ V}$			1	μA
V_{sat}	Saturation voltage (pin 3)	$I = 2 \text{ mA}$			0.4	V
V_{off}	Comparators offset voltage (pins 13, 14, 15)	$V_{\text{ref}} = 1 \text{ V}$			5	mV
I_{c}	Comparator bias current (pins 13, 14, 15)		-100		10	μA
V_{ref}	Input reference voltage (pin 15)		0		3	V
t_{CLK}	Clock time		0.5			μs
t_{S}	Set up time		1			μs
t_{H}	Hold time		4			μs
t_{R}	Reset time		1			μs
t_{RCCLK}	Reset to clock delay		1			μs

Symbol	Parameter	Test conditions	Min.	Typ	Max.	Unit
V_{S}	Supply voltage (pin 12)		4.75		7	V
I_{q}	Quiescent supply current (pin 12)	Outputs floating		50	80	mA
V_{I}	Input voltage (pin 11, 17, 18, 19, 20)	Low			0.6	V
		High	2		V_{S}	V
I_{I}	Input current (pin 11, 17, 18, 19, 20)	$V = \text{L}$		100		μA
		$V = \text{H}$			10	μA
V_{EN}	Enable input voltage (pin 10)	Low			1.3	V
		High	2		V_{S}	V
I_{EN}	Enable input current (pin 10)	$V_{\text{en}} = \text{L}$			100	μA
		$V_{\text{en}} = \text{H}$			10	μA
V_{O}	Phase output voltage (pins 4, 6, 7, 9)	$I_{\text{O}} = 1 \text{ mA}$ V_{OL}			0.4	V
		$I_{\text{O}} = 5 \text{ mA}$ V_{OH}	3.9			V
V_{INH}	Inhibit output voltage (pins 5, 8)	$I_{\text{O}} = 1 \text{ mA}$ $V_{\text{INH L}}$			0.4	V
		$I_{\text{O}} = 5 \text{ mA}$ $V_{\text{INH H}}$	3.9			V
V_{SYNC}	Sync Output Voltage	$I_{\text{O}} = 5 \text{ mA}$ $V_{\text{SYNC H}}$	3.3			V
		$I_{\text{O}} = 5 \text{ mA}$ $V_{\text{SYNC V}}$			0.8	V

รูปที่ 5.8 คุณสมบัติทางไฟฟ้าของไอซี L297

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

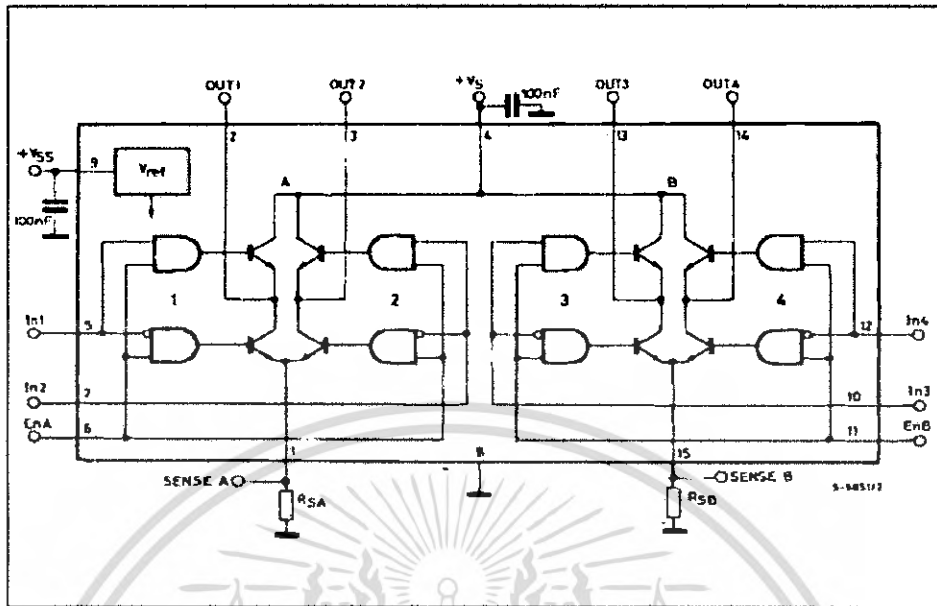


รูปที่ 5.9 แสดงความสัมพันธ์ของสัญญาณ

5.2 รายละเอียดของไอซี L298

- รับแรงดันที่จะจ่ายให้สเต็ปมอเตอร์ได้ถึง 46 V.
- จ่ายกระแสตรงรวมได้ถึง 4 A.
- แรงดันอิมิต์ค่า
- มีระบบป้องกันอุณหภูมิเกิน

โครงสร้างในส่วนของวงจรขับภายในของ L298 เป็นแบบคู่ออต์ฟูลบริดจ์ (Dual Full Bridge) สามารถจ่ายกระแสเอาต์พุตได้สูงสุด 2 แอมป์ต่อเฟส (Phase) ระดับสัญญาณที่ใช้ในการขับ L298 เป็นแบบ TTL เอาต์พุตที่ออกมาสามารถนำไปขับรีเลย์ (Relays), โซลินอยด์ (Solenoid), สเต็ปมอเตอร์ (Steptomotor) ในการขับวงจรภายในไอซีทั้งสองชุดเราสามารถเลือกได้ว่าจะให้ชุดใดชุดหนึ่งหรือทั้งสองชุดทำงานก็ได้ โดยควบคุมที่ขา ENABLE

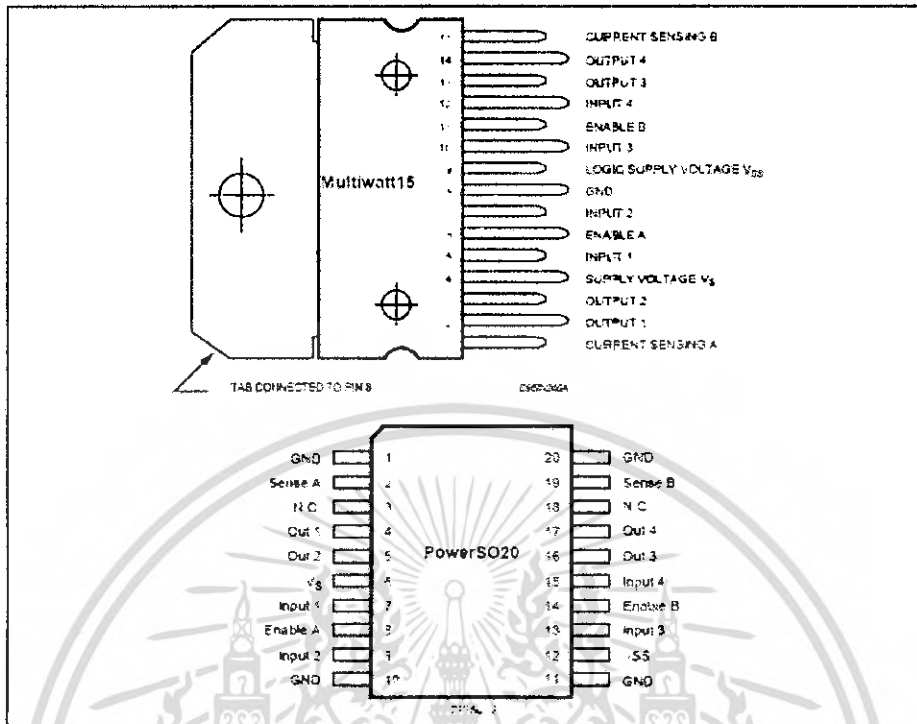


รูปที่ 5.10 โครงสร้างภายในไอซี L298

Symbol	Parameter	Value	Unit
V_S	Power Supply	50	V
V_{SS}	Logic Supply Voltage	7	V
V_i, V_{en}	Input and Enable Voltage	-0.3 to 7	V
I_O	Peak Output Current (each Channel) - Non Repetitive ($t = 100\mu s$) - Repetitive (80% on -20% off $t_{on} = 10ms$) - DC Operation	3 2.5 2	A A A
V_{sens}	Sensing Voltage	-1 to 2.3	V
P_{tot}	Total Power Dissipation ($T_{case} = 75^\circ C$)	25	W
T_{op}	Junction Operating Temperature	-25 to 130	$^\circ C$
T_{stg}, T_j	Storage and Junction Temperature	-40 to 150	$^\circ C$

รูปที่ 5.11 ค่าสูงสุดที่ชี้ในการทำงานของ L298

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.12 ตำแหน่งขาของไอซี L298

MW.15	PowerSO	Name	Function
1;15	2;19	Sense A, Sense B	Between this pin and ground is connected the sense resistor to control the current of the load.
2;3	4;5	Out 1, Out 2	Outputs of the Bridge A; the current that flows through the load connected between these two pins is monitored at pin 1.
4	6	VS	Supply Voltage for the Power Output Stages. A non-inductive 100nF capacitor must be connected between this pin and ground.
5;7	7;9	Input 1, Input 2	TTL Compatible Inputs of the Bridge A.
6;11	8;14	Enable A, Enable B	TTL Compatible Enable Input: the L state disables the bridge A (enable A) and/or the bridge B (enable B).
8	1,10,11,20	GND	Ground.
9	12	VSS	Supply Voltage for the Logic Blocks. A 100nF capacitor must be connected between this pin and ground.
10; 12	13;15	input 3, Input 4	TTL Compatible Inputs of the Bridge B.
13; 14	16;17	Out 3; Out 4	Outputs of the Bridge B. The current that flows through the load connected between these two pins is monitored at pin 15.
-	3;18	N.C.	Not Connected

รูปที่ 5.13 คุณสมบัติของขาต่างของไอซี L298

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

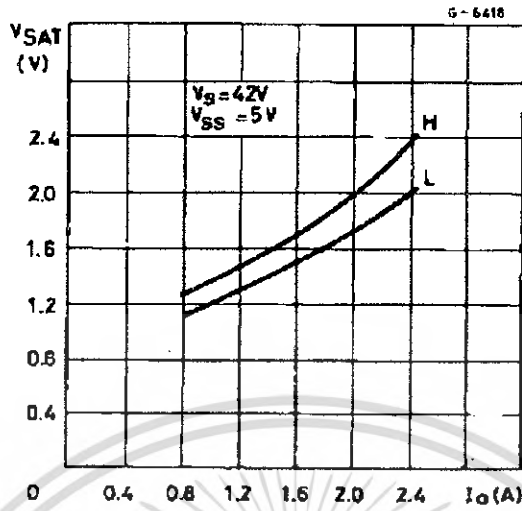
คุณสมบัติทางไฟฟ้า ที่ $V_s = 42V$, $V_{SS} = 5V$, $T_j = 25^\circ C$

Symbol	Parameter	Test Conditions	Min.	Typ.	Max.	Unit
V_S	Supply Voltage (pin 4)	Operative Condition	$V_{I+} + 2.5$		46	V
V_{SS}	Logic Supply Voltage (pin 9)		4.5	5	7	V
I_S	Quiescent Supply Current (pin 4)	$V_{en} = H$; $I_L = 0$ $V_i = L$		13	22	mA
		$V = H$		50	70	mA
I_{SS}	Quiescent Current from V_{SS} (pin 5)	$V_{en} = L$ $V_i = X$		4		mA
		$V_{en} = H$; $I_L = 0$ $V_i = L$		24	36	mA
I_{SS}		$V = H$		7	12	mA
		$V_{en} = L$ $V_i = X$			6	mA
V_{IL}	Input Low Voltage (pins 5, 7, 10, 12)		-0.3		1.5	V
V_{IH}	Input High Voltage (pins 5, 7, 10, 12)		2.3		V_{SS}	V
I_{iL}	Low Voltage Input Current (pins 5, 7, 10, 12)	$V_i = L$			-10	μA
I_{iH}	High Voltage Input Current (pins 5, 7, 10, 12)	$V_i = H \leq V_{SS} - 0.6V$		30	100	μA
$V_{en} = L$	Enable Low Voltage (pins 6, 11)		-0.3		1.5	V
$V_{en} = H$	Enable High Voltage (pins 6, 11)		2.3		V_{SS}	V
$I_{en} = L$	Low Voltage Enable Current (pins 6, 11)	$V_{en} = L$			-10	μA
$I_{en} = H$	High Voltage Enable Current (pins 6, 11)	$V_{en} = H \leq V_{SS} - 0.6V$		30	100	μA
$V_{CEsat(H)}$	Source Saturation Voltage	$I_L = 1A$	0.95	1.35	1.7	V
		$I_L = 2A$		2	2.7	V
$V_{CEsat(L)}$	Sink Saturation Voltage	$I_L = 1A$ (5)	0.85	1.2	1.6	V
		$I_L = 2A$ (5)		1.7	2.3	V
V_{CEsat}	Total Drop	$I_L = 1A$ (5)	1.80		3.2	V
		$I_L = 2A$ (5)			4.5	V
V_{sens}	Sensing Voltage (pins 1, 15)		-1 (1)		2	V

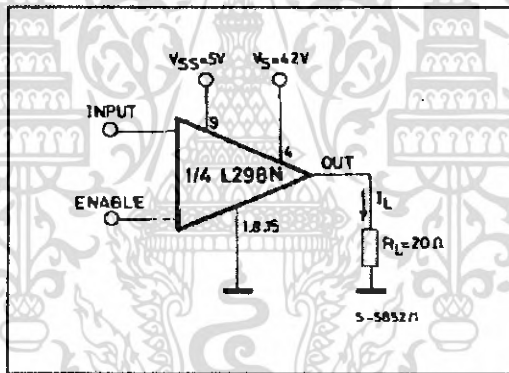
Symbol	Parameter	Test Conditions	Min.	Typ.	Max.	Unit
T_1 (V)	Source Current Turn-off Delay	$0.5V_i$ to $0.9I_L$ (2), (4)		1.5		μs
T_2 (V)	Source Current Fall Time	$0.9I_L$ to $0.1I_L$ (2), (4)		0.2		μs
T_3 (V)	Source Current Turn-on Delay	$0.5V_i$ to $0.1I_L$ (2), (4)		2		μs
T_4 (V)	Source Current Rise Time	$0.1I_L$ to $0.9I_L$ (2), (4)		0.7		μs
T_5 (V)	Sink Current Turn-off Delay	$0.5V_i$ to $0.9I_L$ (3), (4)		0.7		μs
T_6 (V)	Sink Current Fall Time	$0.9I_L$ to $0.1I_L$ (3), (4)		0.25		μs
T_7 (V)	Sink Current Turn-on Delay	$0.5V_i$ to $0.9I_L$ (3), (4)		1.6		μs
T_8 (V)	Sink Current Rise Time	$0.1I_L$ to $0.9I_L$ (3), (4)		0.2		μs
f_c (V)	Commutation Frequency	$I_L = 2A$		25	40	KHz
T_1 (V _{en})	Source Current Turn-off Delay	$0.5V_{en}$ to $0.9I_L$ (2), (4)		3		μs
T_2 (V _{en})	Source Current Fall Time	$0.9I_L$ to $0.1I_L$ (2), (4)		1		μs
T_3 (V _{en})	Source Current Turn-on Delay	$0.5V_{en}$ to $0.1I_L$ (2), (4)		0.3		μs
T_4 (V _{en})	Source Current Rise Time	$0.1I_L$ to $0.9I_L$ (2), (4)		0.4		μs
T_5 (V _{en})	Sink Current Turn-off Delay	$0.5V_{en}$ to $0.9I_L$ (3), (4)		2.2		μs
T_6 (V _{en})	Sink Current Fall Time	$0.9I_L$ to $0.1I_L$ (3), (4)		0.35		μs
T_7 (V _{en})	Sink Current Turn-on Delay	$0.5V_{en}$ to $0.9I_L$ (3), (4)		0.25		μs
T_8 (V _{en})	Sink Current Rise Time	$0.1I_L$ to $0.9I_L$ (3), (4)		0.1		μs

รูปที่ 5.14 คุณสมบัติทางไฟฟ้าของไอซี L298

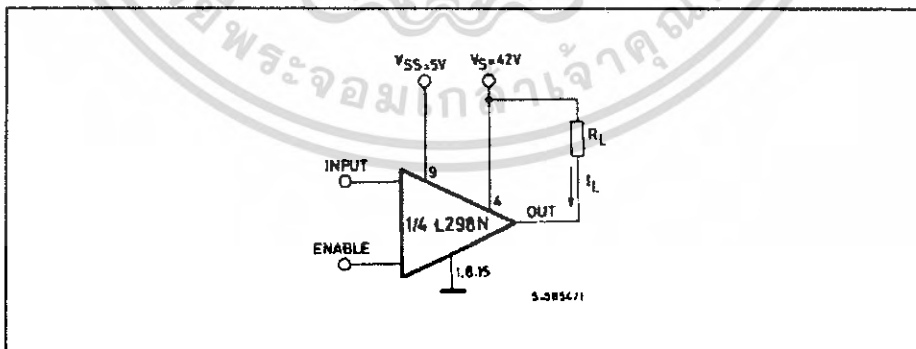
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.15 แสดงค่าแรงดันอิ่มตัวกับกระแสเอาต์พุต



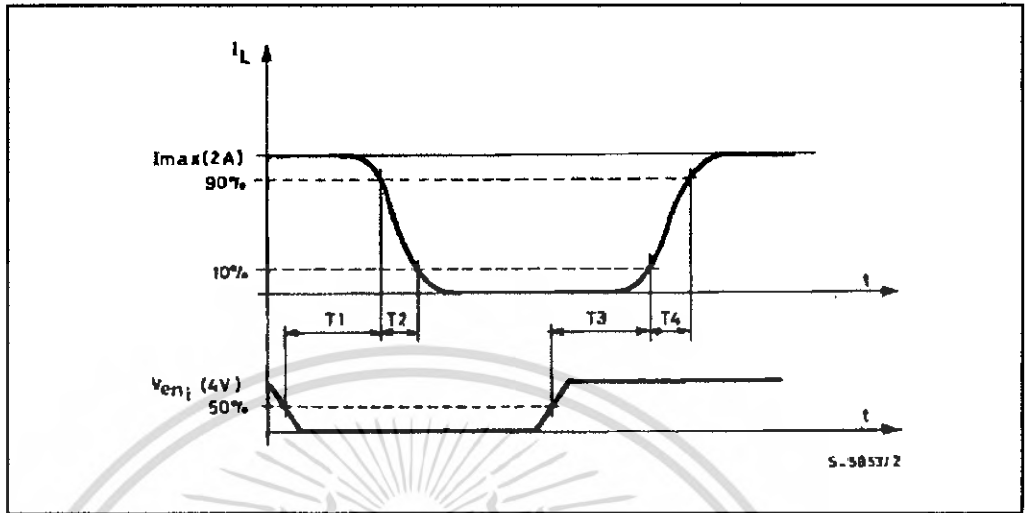
Note : For INPUT Switching, set EN = H
For ENABLE Switching, set IN = H



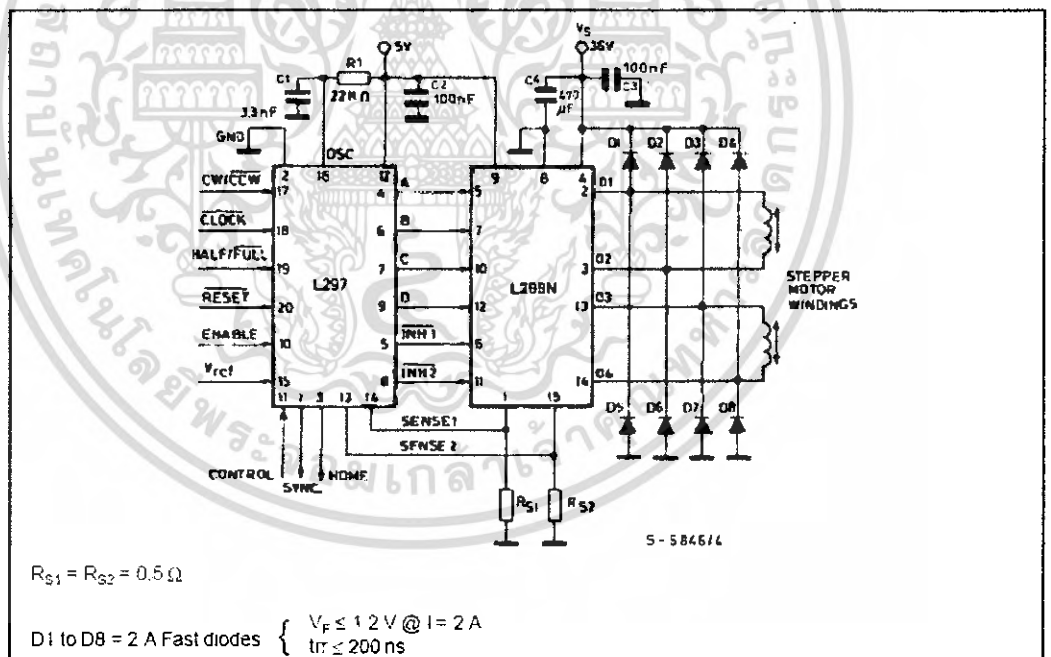
Note : For INPUT Switching, set EN = H
For ENABLE Switching, set IN = L

รูปที่ 5.16 การทดสอบการทำงานเบื้องต้นของไอซี L298

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.17 ค่ากระแส Delay ต่อสัญญาณ Input

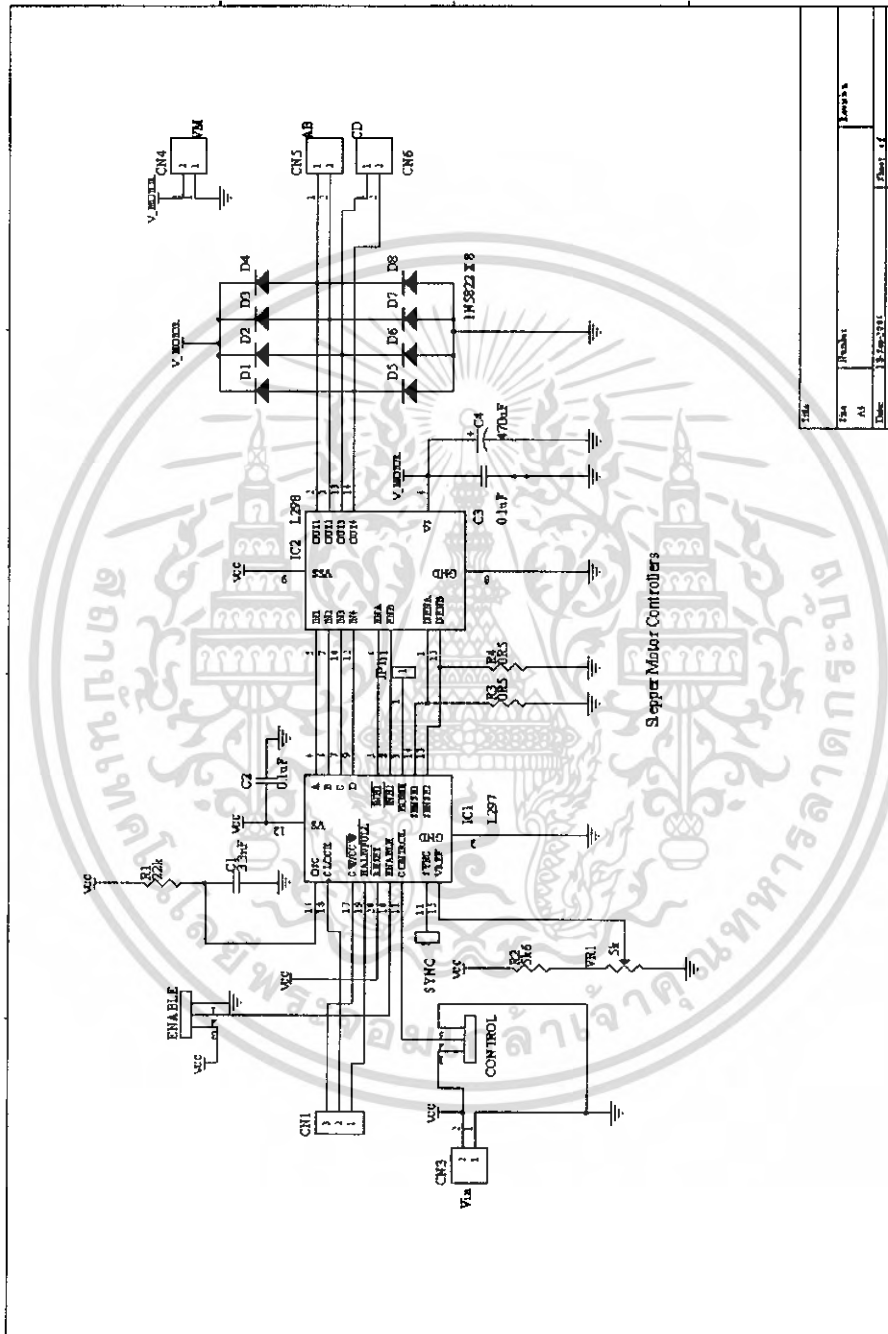


รูปที่ 5.18 การต่อใช้งานระหว่างไอซี L297 กับไอซี L298

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.3 วงจรควบคุมและขับสเต็ปมอเตอร์

ในการออกแบบวงจรควบคุมและขับสเต็ปมอเตอร์ จะใช้ไอซี L297 และ L298 ร่วมกัน แสดงตามรูปที่ 5.19



รูปที่ 5.19 วงจรควบคุมและวงจรถับสเต็ปมอเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

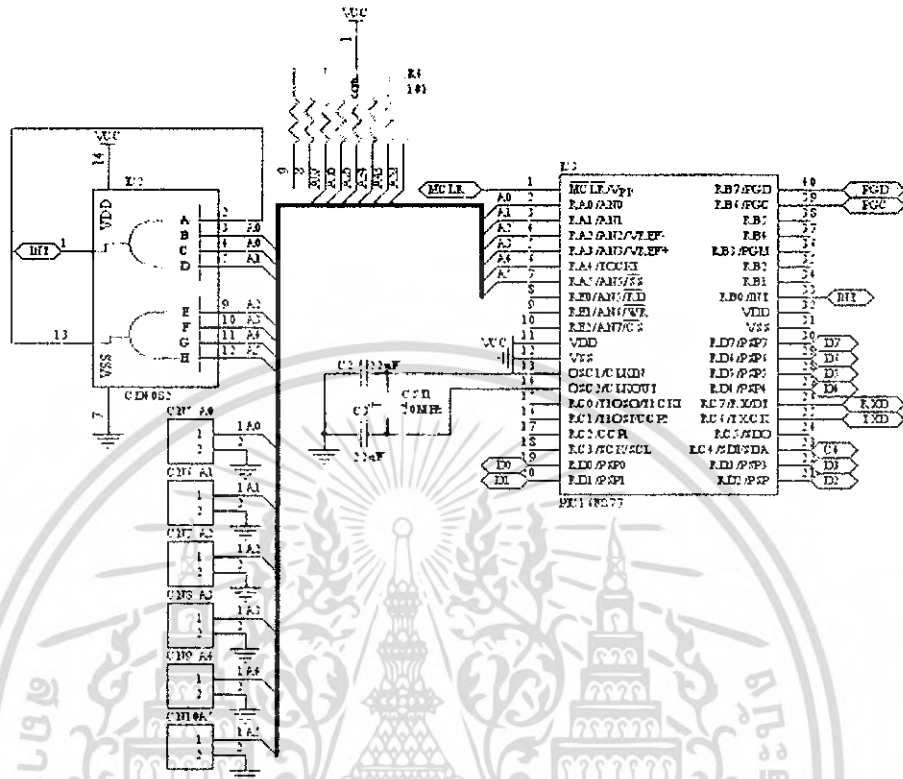
5.3.1 การทำงานของวงจรควบคุมและวงจรขับสเต็ปมอเตอร์

จากรูปที่ 5.19 เป็นวงจรควบคุมและวงจรขับ ซึ่งสามารถขับสเต็ปมอเตอร์ได้เพียงตัวเดียว ซึ่งในการใช้งานจริงจะทำการสร้างทั้งหมด 3 ชุด ในการอธิบายการทำงานจะขออธิบายเพียงชุดเดียวเท่านั้น ส่วนอีกสองชุดจะมีความทำงานเหมือนกัน

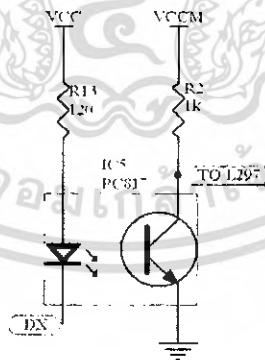
ในการทำงานของ L297 นั้นจะต้องทำการสร้างสัญญาณออสซิลเลต (Oscillate) ให้กับไอซี โดยการต่อตัวต้านทานและตัวเก็บประจุเข้าที่ขา 16 ในการสร้างสัญญาณเอาต์พุตที่จะป้อนให้กับวงจรขับจะต้องทำการป้อนสัญญาณคล็อกให้กับ L297 ที่ขา 18 ในการควบคุมทิศทางการหมุนทำได้โดยการป้อนลอจิกให้กับขา 17 โดยถ้าต้องการให้มอเตอร์หมุนตามเข็มนาฬิกา (CW) ก็ทำการป้อนลอจิกสูง ถ้าต้องการให้มอเตอร์หมุนทวนเข็มนาฬิกา (CCW) ก็ทำการป้อนลอจิกต่ำ ในการควบคุมโหมดการทำงานว่าจะให้ทำงานในโหมดพูลสเต็ปหรือโหมดฮาล์ฟสเต็ปทำได้โดยควบคุมที่ขา 19 โดยถ้าต้องการขับในโหมดพูลสเต็ปก็ทำการป้อนลอจิกต่ำ ถ้าต้องการขับในโหมดฮาล์ฟสเต็ปก็ทำการป้อนลอจิกสูง

การควบคุมว่าจะให้ L297 ทำงานหรือไม่สามารถควบคุมได้จากขาที่ 10 การควบคุมกระแสทางเอาต์พุตจะสัมพันธ์กับ L298 โดยการป้อนแรงดันอ้างอิงเข้าที่ขา 15 โดยค่าแรงดันอ้างอิงนี้จะมีค่าอยู่ระหว่าง 0-2.5 โวลต์ (Volt) โดยการคำนวณหาค่ากระแสเอาต์พุตจะเป็นไปตามสมการ $I_o = V_{ref} / 0.5 \Omega$ โดยค่าความต้านทานค่า 0.5Ω กำหนดได้จากตัวต้านทาน R3,R4 ที่ต่ออยู่ระหว่าง L297 กับ L298 ในการต่อใช้งานทั้งหมด 3 ชุดเราไม่จำเป็นต้องการสร้างสัญญาณออสซิลเลตให้กับ L297 ทุกตัวเราแค่สร้างให้เพียงชุดเดียวเท่านั้นและอีก 2 ชุดที่เหลือเราก็ทำการต่อขาสัญญาณ SYNC เข้ากับ L297 ตัวที่ไม่ได้ใส่ชุดออสซิลเลตให้เท่านั้น ส่วนสัญญาณเอาต์พุตที่ได้จาก L297 ที่ขา 2,3,4,5 นั้นเราจะนำไปต่อให้กับ L298 เพื่อใช้ในการขับ ส่วนสัญญาณที่ออกจากขา 5 กับ 8 ของ L297 นั้นเป็นสัญญาณที่สร้างขึ้นเพื่อที่จะนำไปใช้ควบคุมการทำงานของ L298 ในการจ่ายไฟให้กับวงจรมอเตอร์นั้นเราจะแยกจ่ายให้กับวงจรโดย L297 และ L298 เราจะจ่ายแรงดันคงที่ 5 โวลต์ ในการจ่ายแรงดันให้กับสเต็ปมอเตอร์นั้นก็ขึ้นอยู่กับว่ามอเตอร์นั้นใช้แรงดันและกระแสเท่าไร ในส่วนเอาต์พุตที่ออกจาก L298 ที่จะไปขับสเต็ปมอเตอร์นั้นเราจะต้องต่อไดโอด (Diode) ทรานซิสเตอร์ (Transister) ดูได้จากโครงสร้างภายใน ไว้เพื่อป้องกันแรงดันย้อนกลับเมื่อมอเตอร์หยุดทำงาน ซึ่งเกิดจากการสะสมพลังงานในขดลวดในมอเตอร์ขณะมอเตอร์ทำงาน โดยจะต้องใช้ไดโอดที่ทำงานได้เร็ว โดยเราจะใช้ไดโอดเบอร์ 1N5822 ซึ่งเป็นไดโอดชนิดช็อคกี้ (Schottky Diode)

5.4 วงจรไมโครคอนโทรลเลอร์

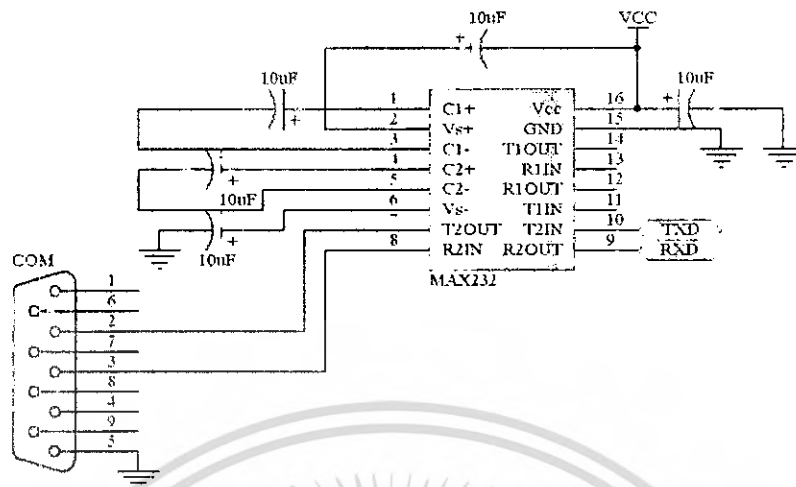


รูปที่ 5.20 วงจร ไมโครคอนโทรลเลอร์ที่ใช้ติดต่อกับคอมพิวเตอร์



รูปที่ 5.21 ส่วนของออปโต้ที่ใช้ในการเชื่อมโยงระหว่างไมโครคอนโทรลเลอร์กับบอร์ดควบคุม และขับเคลื่อนมอเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

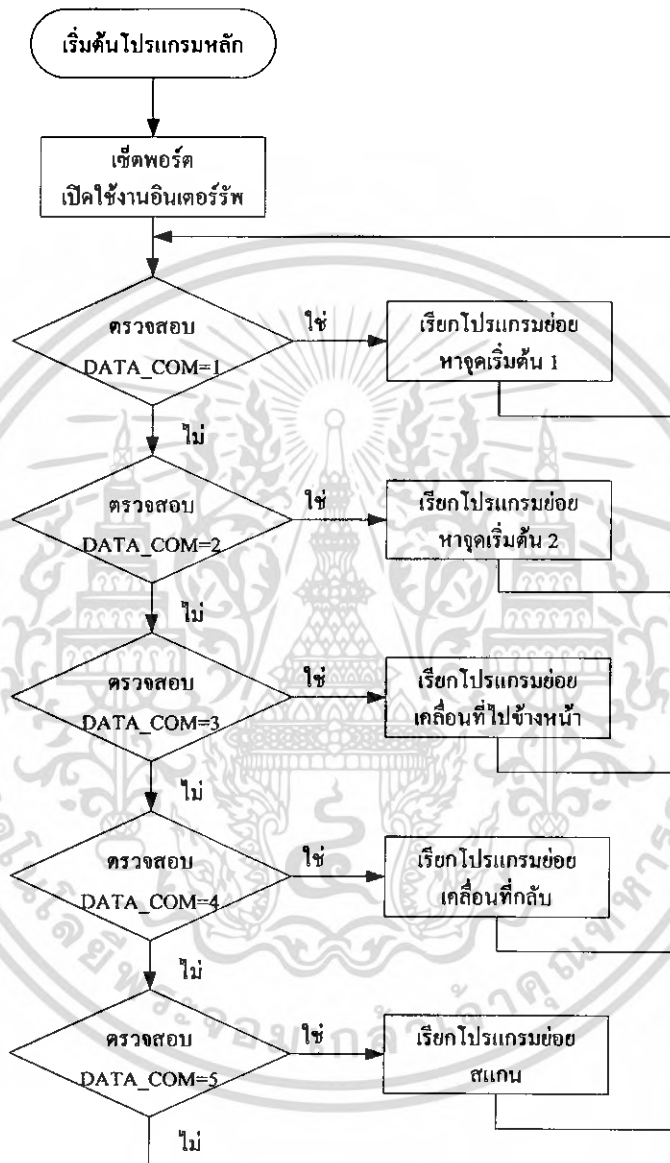


รูปที่ 5.22 ส่วนของวงจรที่ใช้ในการติดต่อระหว่างไมโครคอนโทรลเลอร์กับคอมพิวเตอร์

5.4.1 การทำงานในส่วนของไมโครคอนโทรลเลอร์

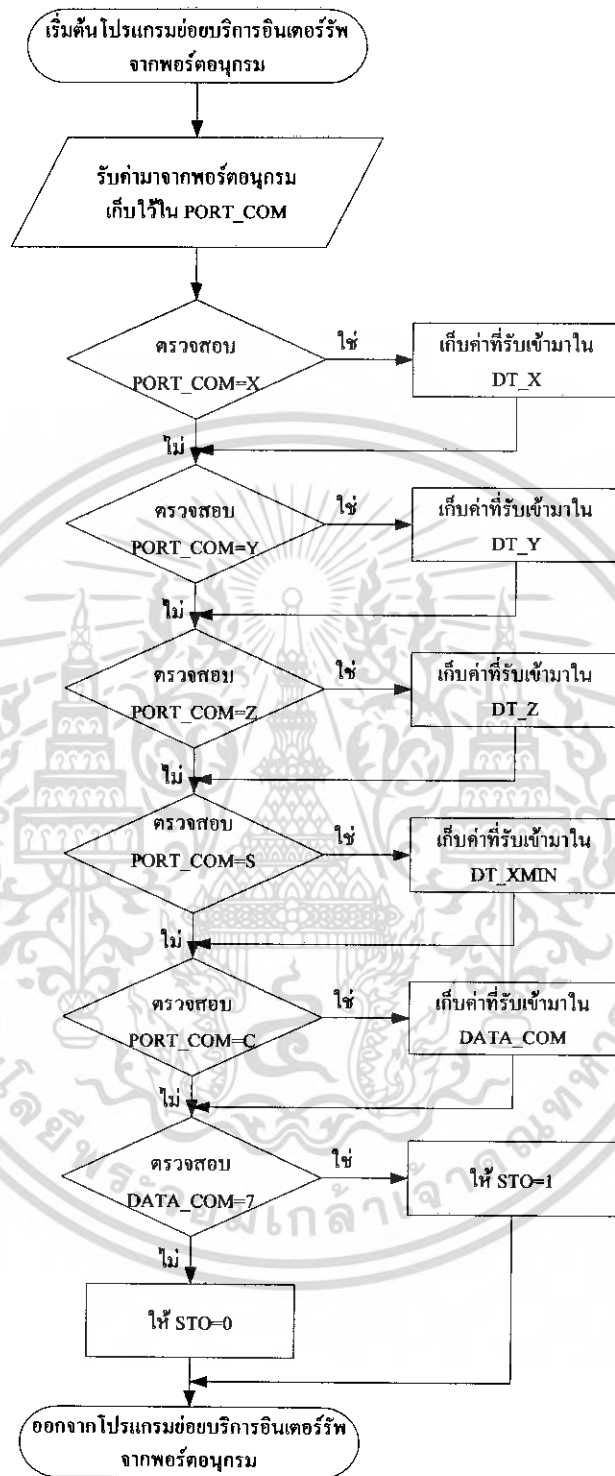
ไมโครคอนโทรลเลอร์จะรับคำสั่งมาจากโปรแกรม LabVIEW ผ่านทางพอร์ตอนุกรมโดยมีไอซี MAX232 ในการปรับระดับสัญญาณ และสัญญาณที่ได้จากไมโครคอนโทรลเลอร์จะทำการส่งไปที่บอร์ดควบคุม และ ขั้วสแต็ปมอเตอร์ ผ่านทางพอร์ต RD โดยมีออปโตคัปเปิลเป็นตัวเชื่อมโยงเพื่อทำการแยกส่วนของไฟสูง และ ไฟต่ำออกจากกัน และมีส่วนของการติดต่อกับลิมิตสวิทช์ เพื่อใช้ในการตรวจสอบการทำงานของแมคคานิค โดยทำการต่อสัญญาณร่วมกันโดยใช้แอนเกทและต่อเข้ากับ ไมโครคอนโทรลเลอร์ผ่านทางพอร์ต RBO เมื่อมีการเปลี่ยนระดับของสัญญาณขึ้นที่ขานี้ก็จะทำให้เกิดอินเตอร์รัปชั่น และจึงทำการวนตรวจสอบอีกที่ว่าเกิดการเปลี่ยนแปลงที่ขาไหน โดยจะทำการต่อลิมิตสวิทช์แต่ละตัวเข้ากับขาของพอร์ต RA

5.5 การออกแบบโปรแกรมควบคุมไมโครคอนโทรลเลอร์



รูปที่ 5.23 โฟลวชาร์ต โปรแกรมหลัก

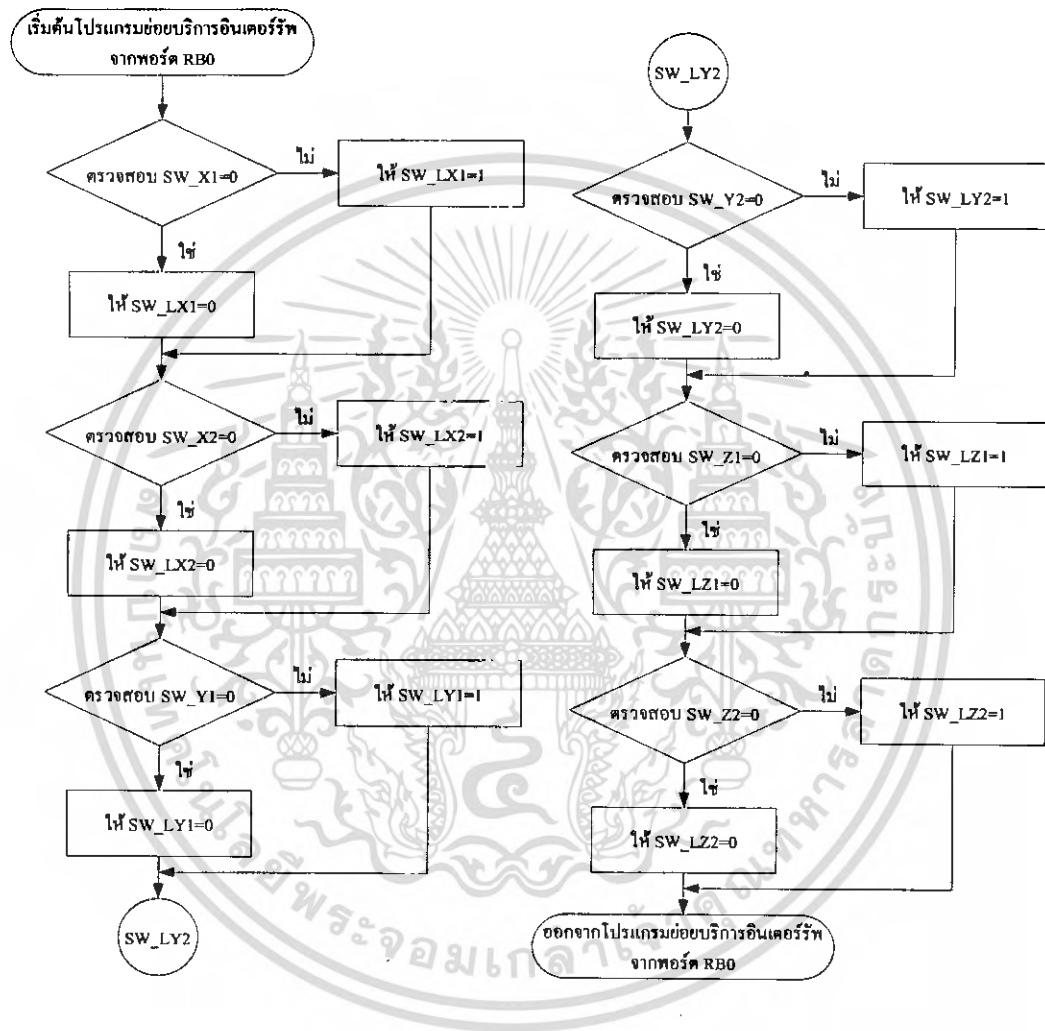
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.24 โฟลวชาร์ตโปรแกรมย่อยบริการอินเทอร์เน็ตจากพอร์ตอนุกรม

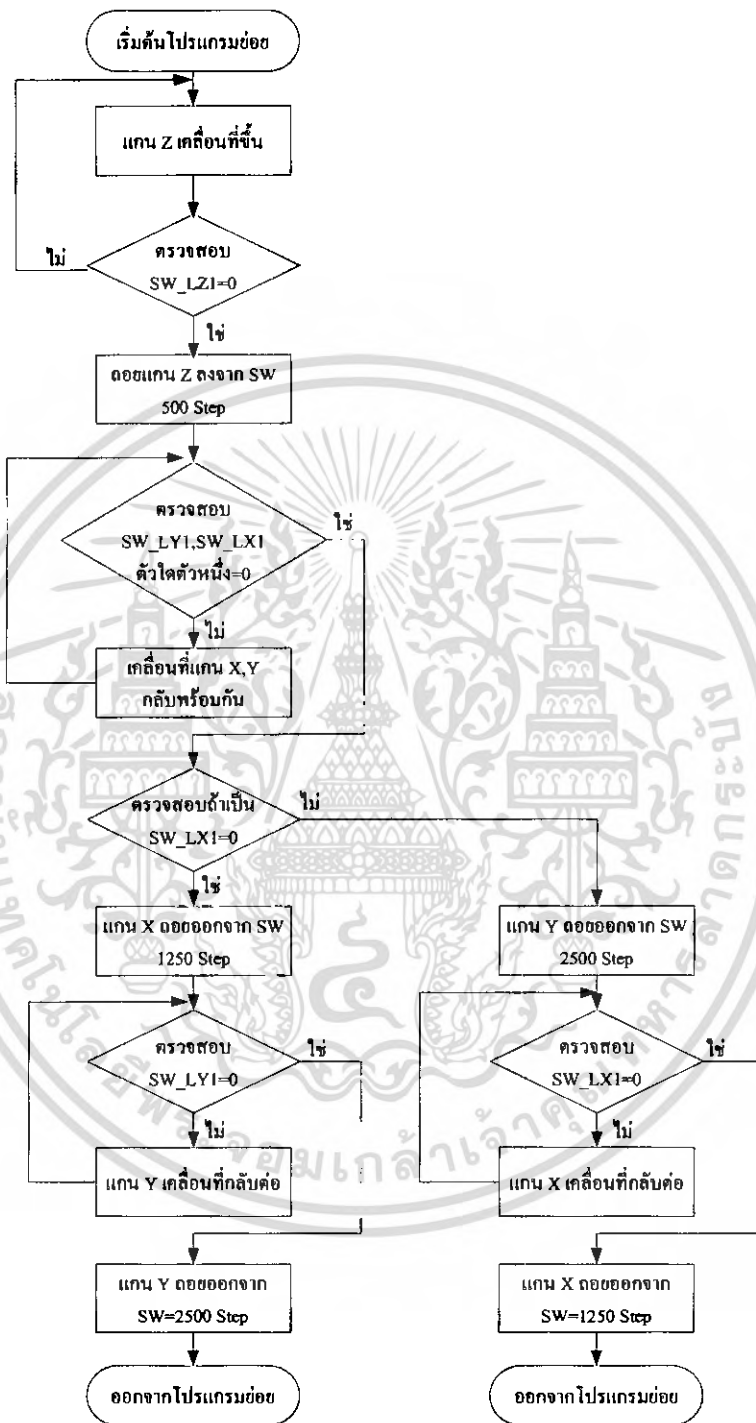
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(แผ่นแทรก ที่ 1)

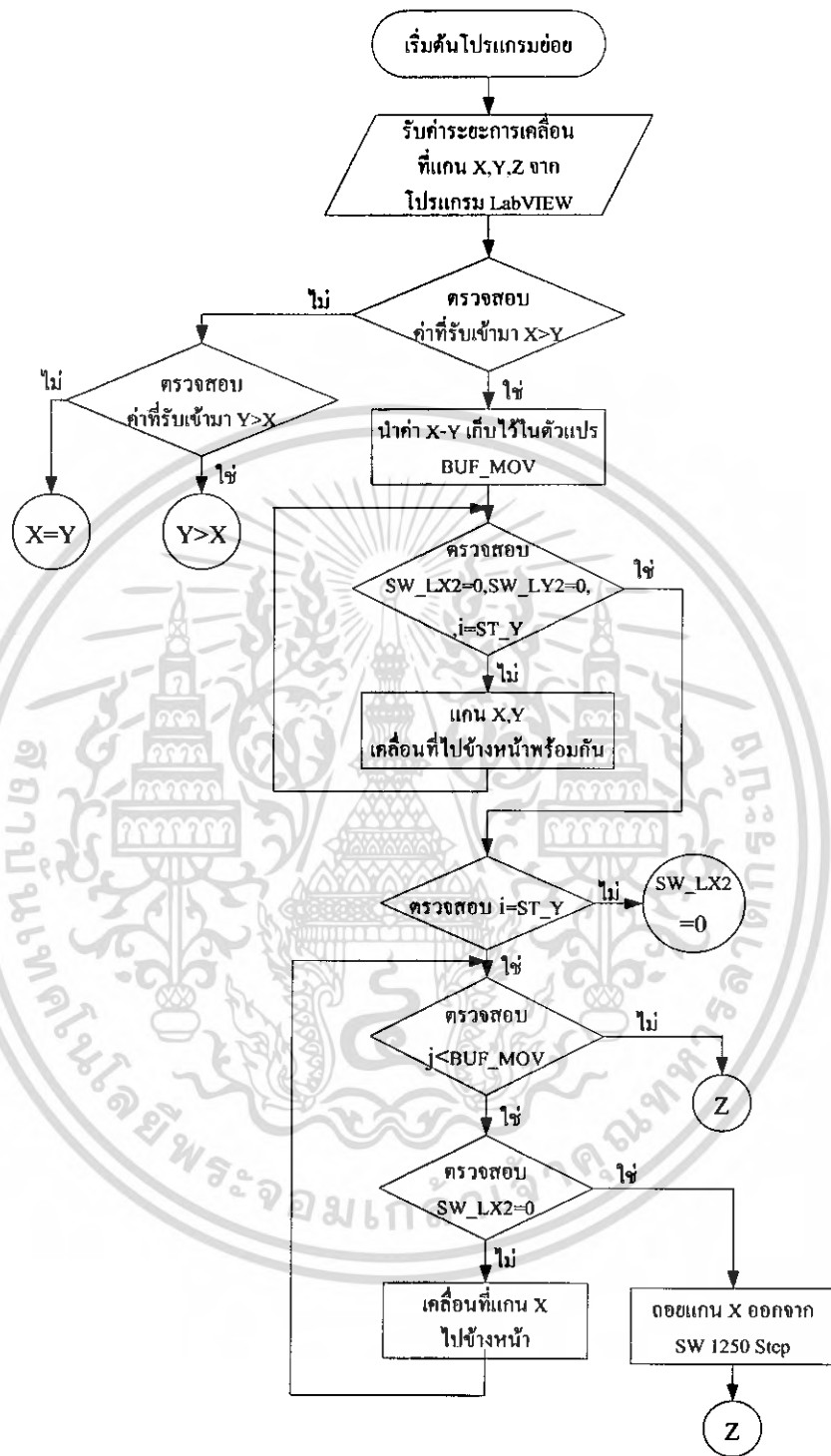


รูปที่ 5.24 ออกจากโปรแกรมย่อยบริการอินเทอร์เน็ตจากพอร์ต RB0 (ต่อ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

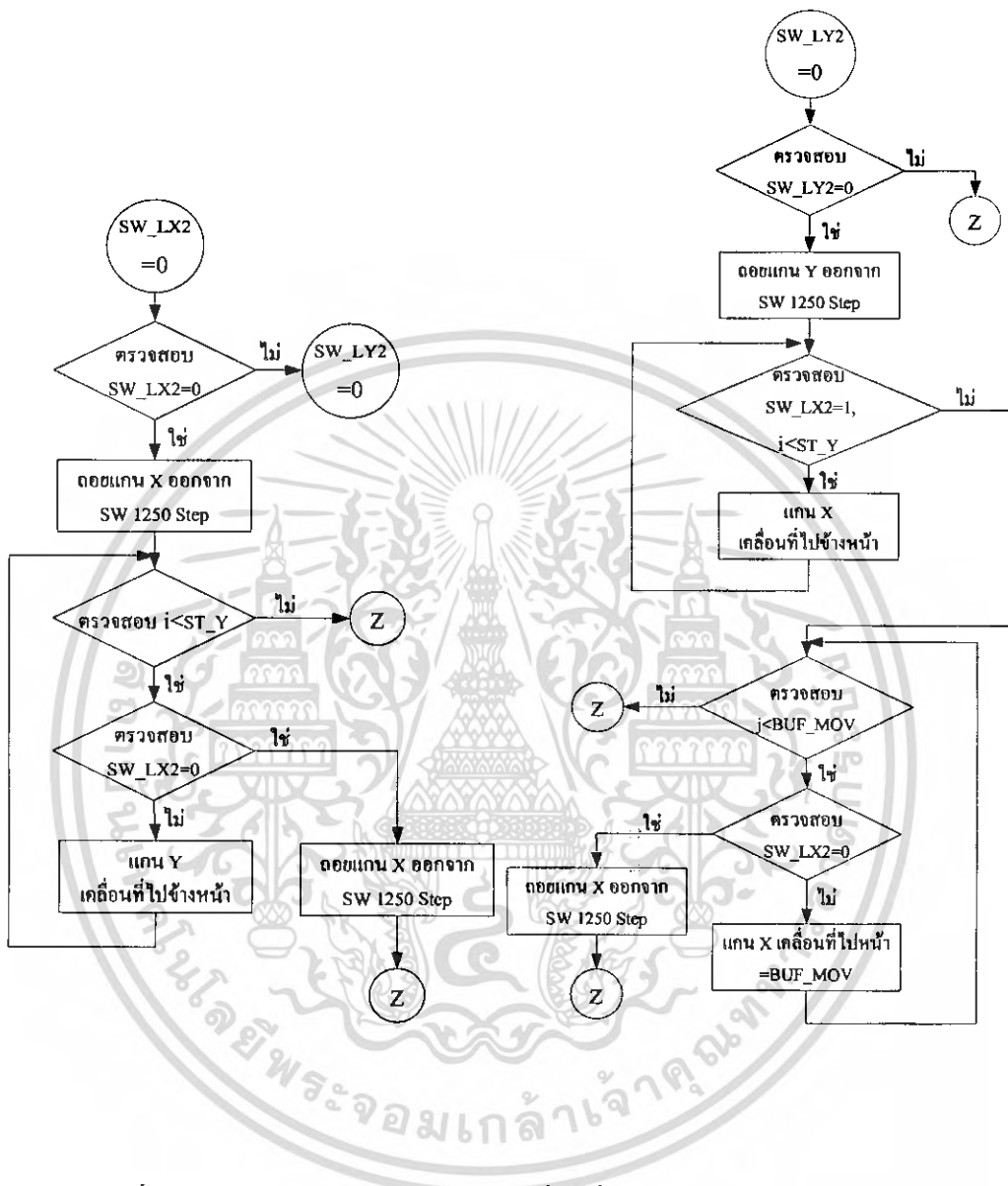


รูปที่ 5.25 โฟลวชาร์ต โปรแกรมย่อยการหาจุดเริ่มต้น



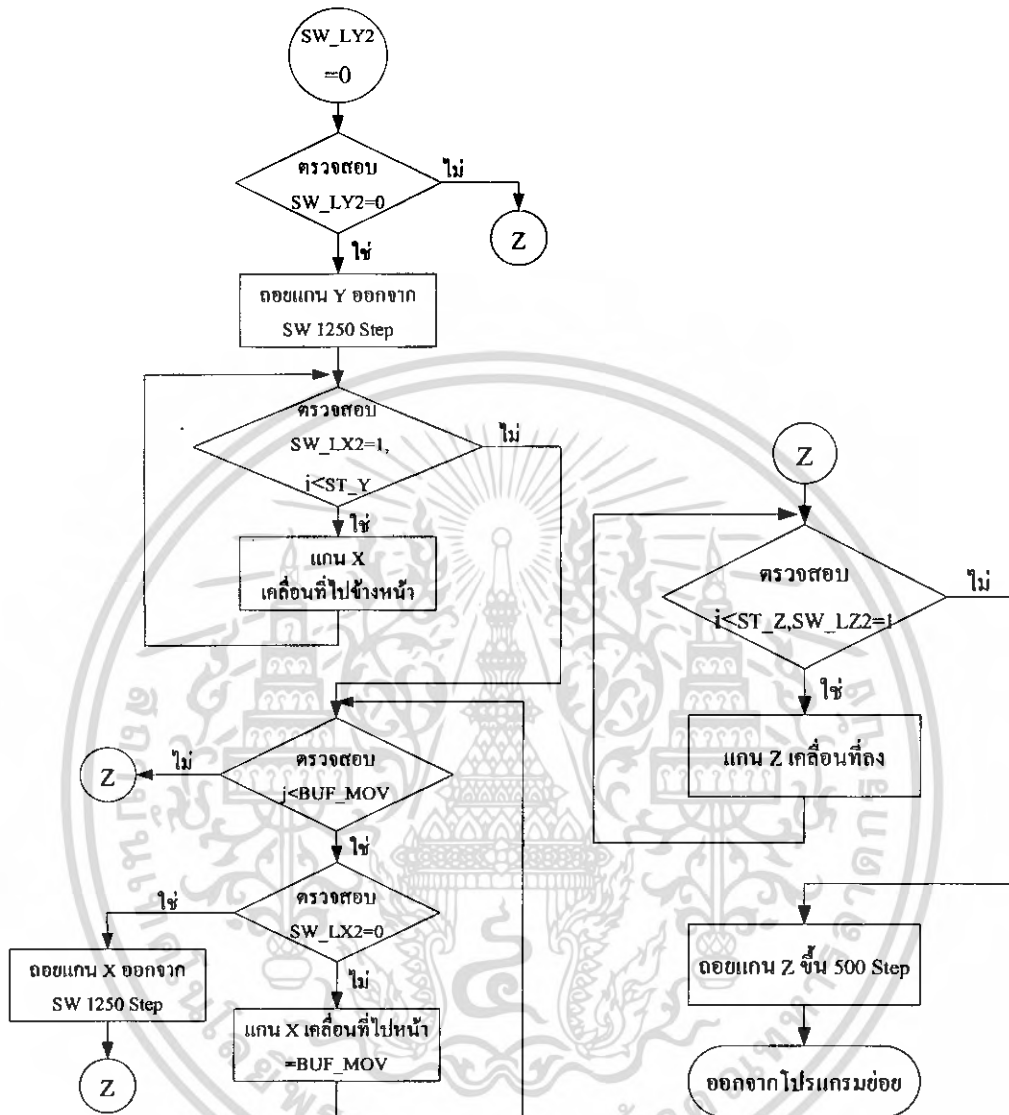
รูปที่ 5.26 โฟลวชาร์ตโปรแกรมย่อยการเคลื่อนที่แกน X,Y,Z ไปข้างหน้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

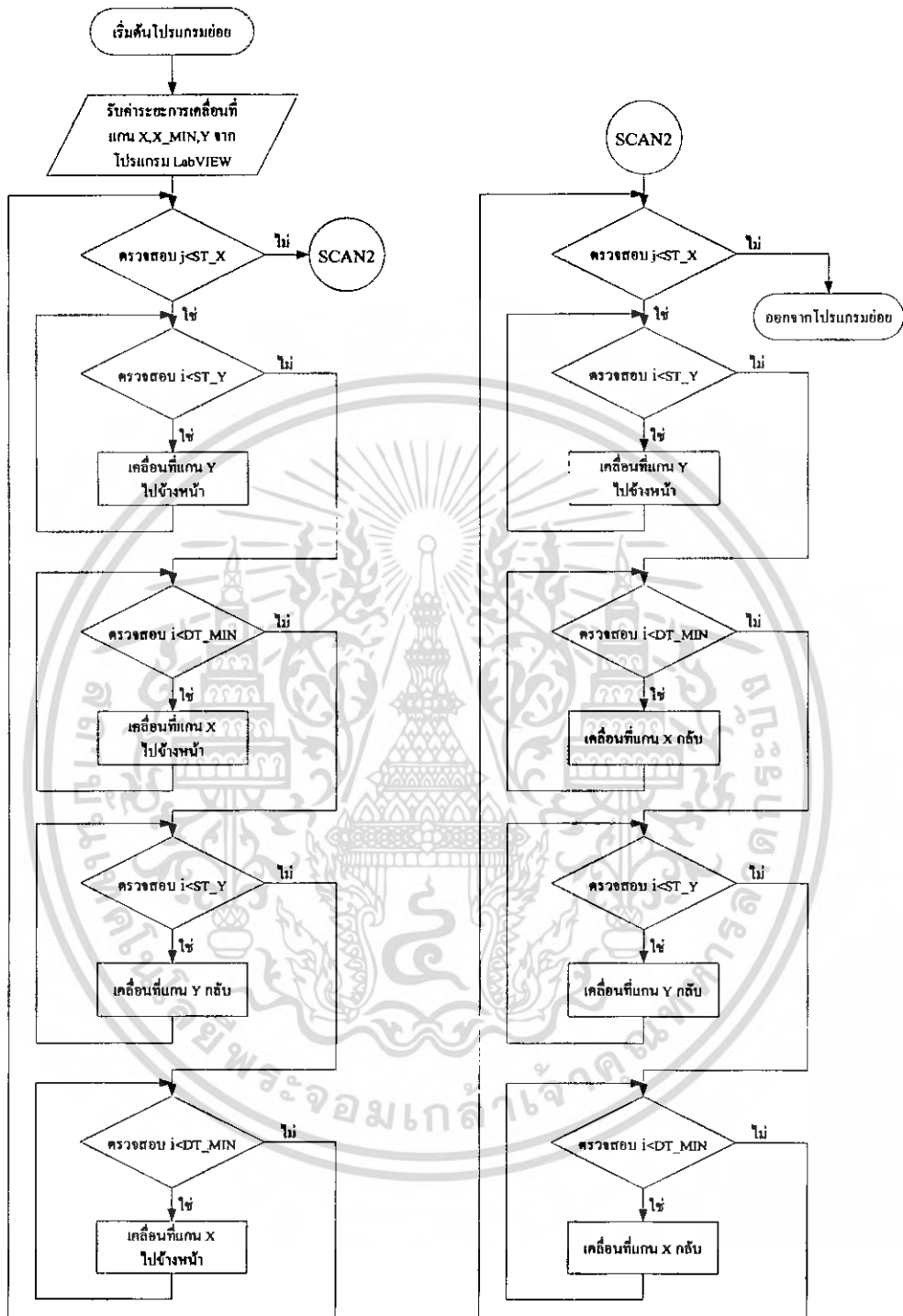


รูปที่ 5.27 โฟลวชาร์ตโปรแกรมข้อยการเคลื่อนที่แกน X,Y,Z ไปข้างหน้า (ต่อ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.28 โฟลวชาร์ต โปรแกรมย่อยการเคลื่อนที่แกน X,Y,Z ไปข้างหน้า (ต่อ)



รูปที่ 5.29 โฟลวชาร์ตโปรแกรมย่อยการสแกน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.5.1 การทำงานในส่วนของโปรแกรม

การทำงานของโปรแกรมควบคุมไมโครคอนโทรลเลอร์เราจะใช้โปรแกรม CCS Compiler ในการคอมไพล์ (Compile) โดยจะเป็นการใช้ภาษาซีในการเขียนโปรแกรมควบคุม จะเริ่มจากส่วนของโปรแกรมหลักทำการเซ็ตค่าต่าง ๆ ซึ่งจะมีส่วนที่สำคัญอยู่คือในส่วนของการเปิดใช้งาน การเกิดอินเทอร์รัพ จากการเปลี่ยนระดับลอจิกที่ขาพอร์ต RB0 และ จากการรับส่งข้อมูลของพอร์ตอนุกรม และก็เป็นส่วนของเงื่อนไขในการเข้าไปทำในส่วนของโปรแกรมห่อย ซึ่งจะทำการวนรอรับค่าอยู่ตลอดเวลา ซึ่งค่าของเงื่อนไข จะได้มาจากการเกิดอินเทอร์รัพ ทั้งหมด

โดยเมื่อเกิดการอินเทอร์รัพขึ้นที่โปรแกรมห่อยบริการอินเทอร์รัพจากพอร์ตอนุกรม เราจะใช้คำสั่ง `getc()` ในการรับค่าเข้ามาในชุดแรก ซึ่งจะถูกส่งออกมาจากโปรแกรม LabVIEW ค่าตัวแรกที่รับเข้ามานี้จะเป็นค่าตัวอักษร ซึ่งเมื่อรับเข้ามาก็จะทำการตรวจสอบเงื่อนไขว่าตรงกับเงื่อนไขใด ค่าตัวต่อไปที่จะรับเข้ามาก็จะนำไปเก็บที่เงื่อนไขนั้น โดยค่าที่รับเข้ามาเก็บในเงื่อนไขนั้น หลักๆจะมีอยู่สองประเภทคือ ค่าที่เป็นอักษรกับค่าที่เป็นข้อมูลเลขฐานสิบหก โดยค่าเลขฐานสิบหกนั้นเราจะใช้คำสั่ง `gethex()` ในการรับค่าเข้ามาเก็บ และจะนำไปใช้ในการ ส่งค่าให้กับฟังก์ชัน (Function) เพื่อกำหนดระยะเวลาเคลื่อนที่ของแกนต่างๆ และถ้าเป็นค่าตัวอักษรก็จะนำไปใช้ในส่วนของโปรแกรมหลัก ซึ่งจะวนรอรับค่าอักษรอยู่ว่า ตรงกับเงื่อนไขใดก็จะทำการเรียกฟังก์ชันย่อย และ ถ้าฟังก์ชันนั้นมีการเรียกรับค่า ก็จะส่งค่าให้กับฟังก์ชันนั้น

ส่วนของโปรแกรมห่อยบริการอินเทอร์รัพที่เกิดจากการเปลี่ยนระดับลอจิกที่ขาพอร์ต RB0 นั้นเมื่อเกิดอินเทอร์รัพขึ้น โปรแกรมก็จะทำการตรวจสอบว่าเกิดการอินเทอร์รัพมาจากขาพอร์ตใด จากนั้นก็ จะไปเข้าเงื่อนไขกำหนดค่าให้กับตัวแปรในเงื่อนไขนั้น โดยค่าตัวแปรในเงื่อนไขนั้นจะนำไปเป็น เงื่อนไขในการทำงานส่วนของโปรแกรมห่อยต่าง ๆ เพื่อให้ตรวจสอบ การเคลื่อนที่ของชุดแม่คานิดในกรณีหาจุดเริ่มต้น และ กรณีการเคลื่อนที่เกินกำหนด

โดยในส่วนของฟังก์ชันย่อยหลัก ๆ จะมีอยู่ 4 ฟังก์ชันด้วยกันประกอบด้วย ฟังก์ชันหาจุดเริ่มต้น (origin) ฟังก์ชันเคลื่อนที่แต่ละแกนไปข้างหน้า (`move_forward`) ฟังก์ชันเคลื่อนที่แต่ละแกนกลับ (`move_backward`) โดยการเคลื่อนที่กลับจะมีค่าตัวแปรเหมือนกับการเคลื่อนที่ไปข้างหน้า แต่ค่าตัวแปรทิศทางการเคลื่อนที่จะตรงข้ามกับการเคลื่อนที่ไปข้างหน้า และ ฟังก์ชันการสแกน (`scan`)

บทที่ 6

กลไกการขับเคลื่อนของเครื่องสแกน 3 มิติ

เครื่องสแกน 3 มิติ จะถูกควบคุมด้วยการเขียน โปรแกรมควบคุมการทำงานและสั่งงานของเครื่องสแกน 3 มิติ ลงบน โปรแกรม LabVIEW 7.0 ติดต่อกับไมโครคอนโทรลเลอร์ผ่านทางพอร์ตอนุกรม โดยที่จะสามารถควบคุมความเร็วของ สเต็ปมอเตอร์ ทั้ง 3 แกนได้ ตามต้องการ ไม่ว่าจะเป็นเรื่องของการกำหนดขนาด, ระยะ, ตำแหน่ง เพื่อให้ทำงานตามแบบที่ดั่งโดยการสั่งงานจะส่งผ่านคอมพิวเตอร์ให้เครื่องสแกน 3 มิติ เริ่มทำงาน จนกระทั่ง สิ้นสุดการทำงาน

6.1 หลักการโดยทั่วไปในการออกแบบส่วนประกอบต่างๆ

จุดประสงค์ในการออกแบบการใช้งานของ โครงงานนี้เพื่อ ใช้ในการทำหน้าที่เลื่อนตำแหน่งของหัวจับอัลตราโซนิก ในการสแกนหารอย Crack ของแท่งโลหะ

ออกแบบ โครงสร้างของเครื่องสแกน 3 มิติ รูปแบบการเคลื่อนที่ของแต่ละแกน ขนาดพื้นที่ใช้งาน (Working area), ภาระการทำงานของเครื่อง (load)

- จำนวนเพื่อหาขนาดที่จะเลือกใช้วัสดุ อุปกรณ์ ต่างๆ ไม่ว่าจะเป็น
- การเลือกขนาดมอเตอร์
- การเลือกขนาด BallScrew
- การเลือกใช้ขนาด Linear
- ความแม่นยำ, ความละเอียด และความทนทานทั้งด้าน โครงสร้าง ,การทำงานที่ต่อเนื่อง
- งบประมาณ

6.2 ส่วนประกอบสำคัญในการสร้างเครื่องสแกน 3 มิติ

- วัสดุที่เลือกมาทำโครงสร้าง เหล็กหล่อ และ Aluminium ขึ้นอยู่กับลักษณะงานว่างานหนักหรืองานเบา งาน เจาะปรี้นท์อาจเลือกใช้ Aluminium Profile เป็นโครงสร้าง แต่ราคาจะสูง
- ชุดขับเคลื่อนการเคลื่อนที่ของเครื่องจักร
- ชุดขับเคลื่อน มีทั้ง Feed Screw และ Ball Screw เลือกตามลักษณะงาน ความละเอียดและความแม่นยำ
- ชุดรองเลื่อน ที่กล่าวถึงคือ Linear Guide Way หรือ สไลด์
- ชุดควบคุมการขับเคลื่อน (Driver Board) ทำหน้าที่ควบคุม Stepper Motor ทั้ง 3 ตัว ในแต่ละแกน
- โปรแกรมควบคุมการทำงาน (Software) โครงการนี้ใช้โปรแกรม LabVIEW 7.0 เป็นตัวควบคุมบอร์ดไมโครคอนโทรลเลอร์ PIC ผ่านทางพอร์ตอนุกรม

6.3 การทำงานของเครื่องสแกน 3 มิติ

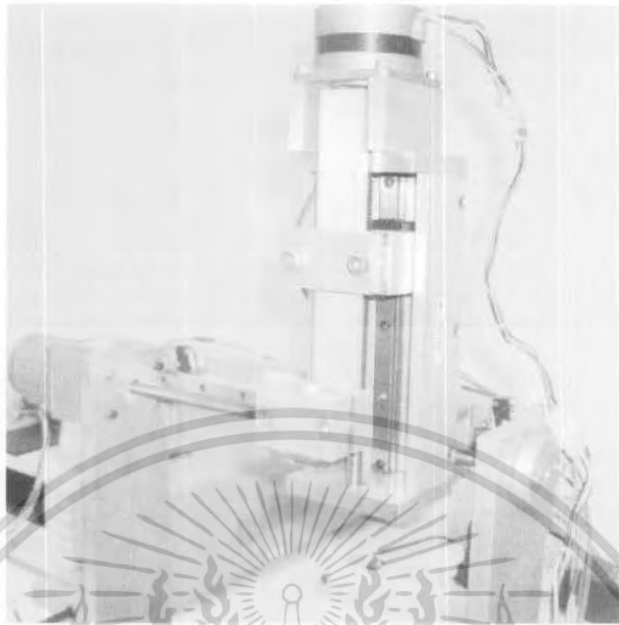
สามารถนำไปใช้งานได้เมื่อเปลี่ยนแปลงข้อมูลให้อยู่ในรูปของ G-Code โดยใช้ CAD-CAM Software เพื่อควบคุมการทำงานของแกน X, Y และ Z โดยผ่านทาง Step Motor Driver Board X, Y และ Z หรือ Servo Motor ให้สามารถควบคุมตำแหน่งและทิศทาง การเคลื่อนที่ของแกนต่างๆ เพื่อได้ตามชิ้นงานนั้นๆ

6.4 แสดงรูปโครงสร้างส่วนต่างๆของระบบ (Mechanic)



รูปที่ 6.1 ชุดเคลื่อนที่ทั้ง 3 แกน เมื่อซิดติดพร้อมกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6.2 ชุดขับเคลื่อนแกน Z



รูปที่ 6.3 แสดงชุดขับเคลื่อนทั้ง 3 แกน ของเครื่องสแกนระบบ 3 มิติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 7

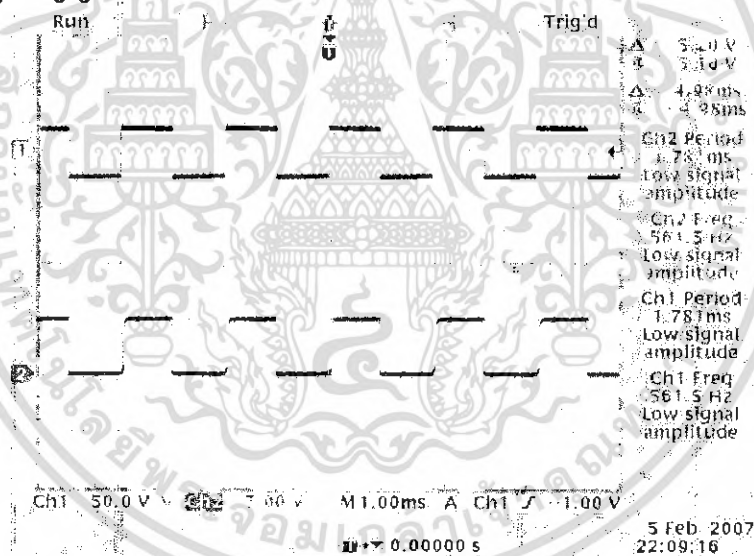
ผลการทดลอง

ผลการทดลองแบ่งออกเป็น 4 ส่วน คือ

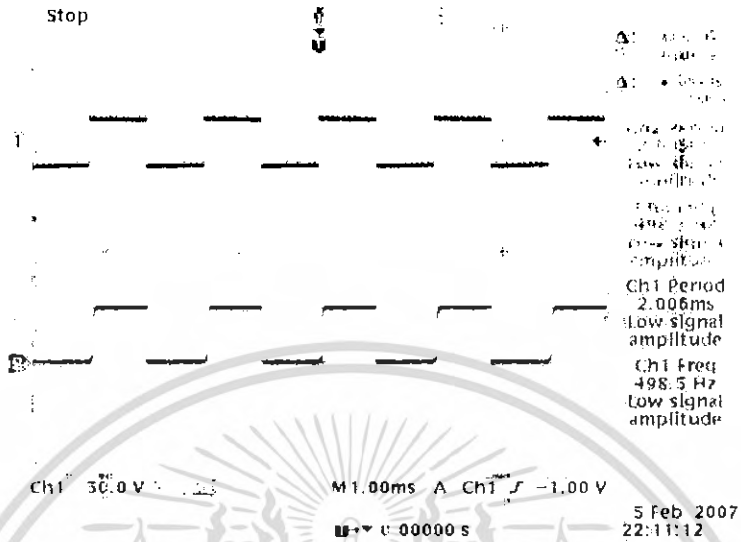
1. การวิเคราะห์สัญญาณจากวงจรไมโครคอนโทรลเลอร์และจากวงจรไครฟมอเตอร์
2. การหาระยะในการเคลื่อนที่ ต่อการจับสัญญาณ 1 STEP
3. การเขียนโปรแกรม LabVIEW
4. Source Code สำหรับไมโครคอนโทรลเลอร์

ผลการทดลอง

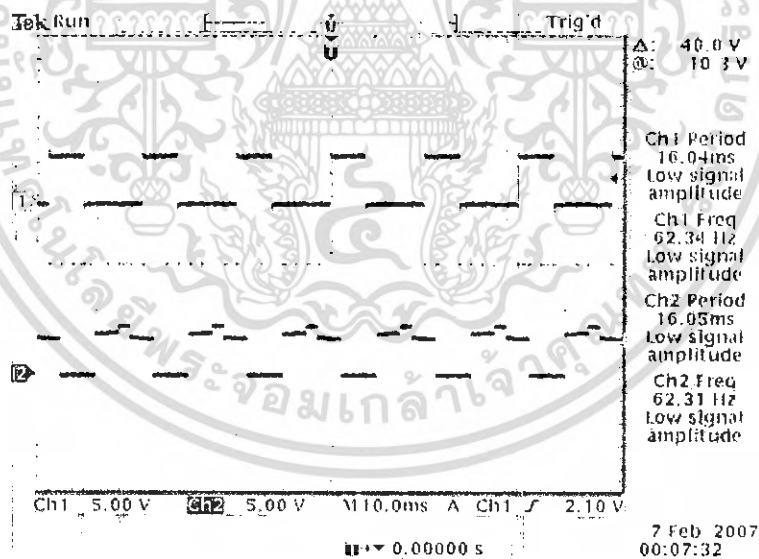
7.1 การวิเคราะห์สัญญาณจากวงจรไมโครคอนโทรลเลอร์และวงจรไครฟมอเตอร์



รูปที่ 7.1 CH1 สัญญาณที่ออกมาจากไมโครคอนโทรลเลอร์ CH2 สัญญาณที่ออกมาจากออปโตที่ คาบเวลาเท่ากับ 1.8 ms. คิวซีไซเคิลเท่ากับ 50%

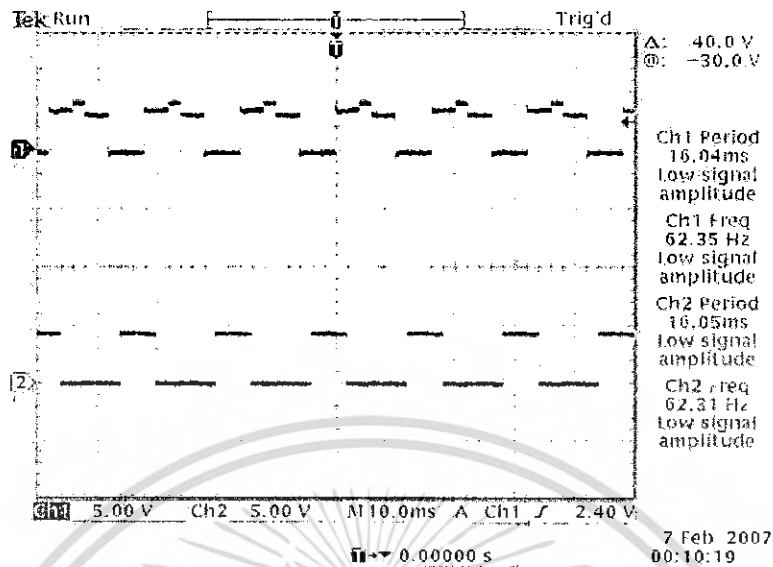


รูปที่ 7.2 CH1 สัญญาณที่ออกมาจากไมโครคอนโทรลเลอร์ CH2 สัญญาณที่ออกมาจากอปโตไดร์
คาบเวลาเท่ากับ 2 ms. คิวดีไซเคิลเท่ากับ 50%

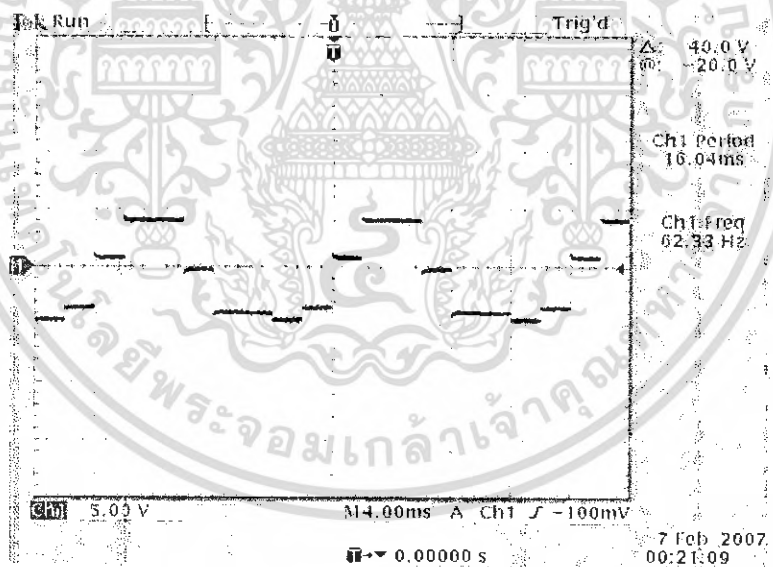


รูปที่ 7.3 CH1 สัญญาณเฟส A และ CH2 สัญญาณเฟส B ที่ออกมาจากไอซี L297 เมื่อป้อนคล็อก
คาบเวลาเท่ากับ 2 ms. คิวดีไซเคิลเท่ากับ 50%

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

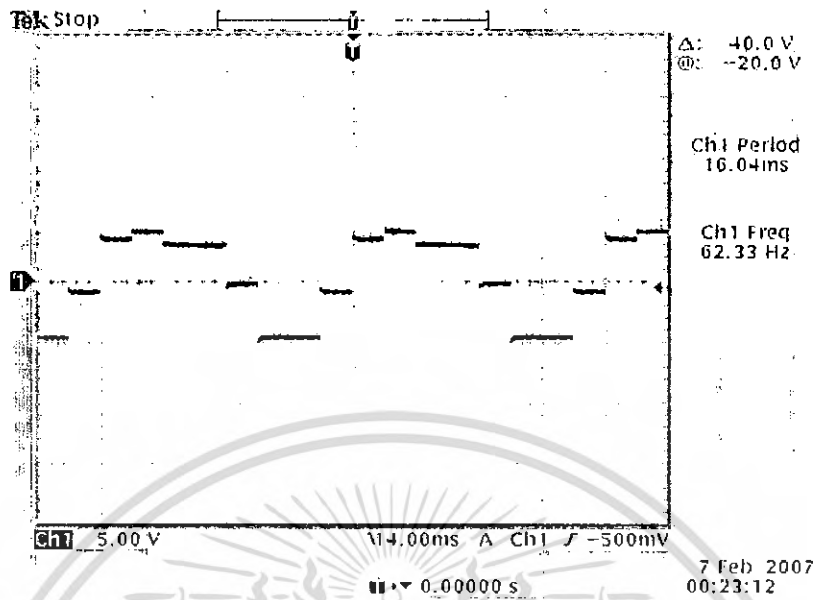


รูปที่ 7.4 CH1 สัญญาณเฟส C และ CH2 สัญญาณเฟส D ที่ออกมาจากไอซี L297 เมื่อป้อนคลิก
คาบเวลาเท่ากับ 2 ms. คิวตี้ไซเคิลเท่ากับ 50%



รูปที่ 7.5 สัญญาณที่ได้จากไอซี L297 เมื่อวัดระหว่างเฟส A กับ B ร่วมกัน เมื่อป้อนคลิกคาบเวลา
เท่ากับ 2 ms. คิวตี้ไซเคิลเท่ากับ 50%

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 7.6 สัญญาณที่ได้จากไอซี L297 เมื่อวัดระหว่างเฟส C กับ D ร่วมกัน เมื่อป้อนคลื่นคาบเวลาเท่ากับ 2 ms. คิวดีไซเคิลเท่ากับ 50%

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7.2 การหาระยะในการเคลื่อนที่ ต่อการขับสัญญาณ 1 STEP

จำนวน STEP	ระยะที่วัดได้ (Cm)											AVERAGE (Cm)	ระยะที่ได้ ต่อ 1 STEP	μm
2,000	0.85	0.80	0.80	0.80	0.80	0.80	0.80	0.80	0.85	0.80	0.80	0.81	4.05	μm
4,000	1.50	1.60	1.50	1.60	1.60	1.60	1.60	1.60	1.60	1.60	1.60	1.58	3.95	μm
4,000	1.50	1.60	1.60	1.60	1.60	1.60	1.60	1.60	1.60	1.60	1.60	1.59	3.97	μm
4,000	1.60	1.60	1.50	1.50	1.60	1.60	1.55	1.60	1.60	1.60	1.60	1.57	3.93	μm
8,000	3.00	3.50	3.00	3.00	3.00	3.50	3.00	3.00	3.50	3.00	3.00	3.15	3.93	μm
10,000	4.00	4.25	4.00	4.00	4.25	4.00	4.00	3.55	4.00	4.00	4.00	4.00	4.00	μm
12,500	5.00	5.00	5.25	5.00	5.00	5.25	5.00	5.00	5.00	5.00	5.00	5.05	4.04	μm
17,500	7.00	6.55	7.00	7.00	7.50	6.55	7.00	7.00	7.00	7.00	7.00	6.96	3.97	μm
22,500	9.00	9.25	9.00	9.00	9.00	9.00	9.00	9.50	9.00	9.00	9.00	9.07	4.03	μm
25,000	10.00	10.50	10.00	10.00	10.00	10.50	10.00	10.00	10.00	10.00	10.50	10.15	4.06	μm
50,000	20.00	20.50	20.00	20.00	20.00	20.50	20.00	20.00	20.50	20.00	20.00	20.15	4.03	μm
60,000	24.00	24.00	24.00	24.00	24.20	24.00	24.00	24.20	23.55	24.00	24.00	23.99	3.99	μm
65,000	26.00	26.50	26.00	26.00	26.50	26.00	26.00	26.00	26.00	26.00	26.00	26.10	4.01	μm

AVE 4.00 μm

ตารางที่ 7.1 การหาระยะการเคลื่อนที่ ต่อการขับสัญญาณ 1 STEP

จากการทดลองหาระยะการเคลื่อนที่ โดยเราต้องการทราบว่าใน 1 STEP จะเคลื่อนที่ได้ระยะเท่าไร โดยทำการป้อนค่าจำนวน STEP ที่แตกต่างกัน 13 ค่า แต่ละค่าจะทำการวัดระยะการเคลื่อนที่จำนวน 10 ครั้ง จากนั้นนำค่าเฉลี่ยของแต่ละครั้งมาหาค่าเฉลี่ยอีกที ซึ่งจากการทดลองเราได้ผลของระยะการเคลื่อนที่ต่อ 1 STEP เท่ากับ 0.4 μm โดยจะนำค่าระยะการเคลื่อนที่นี้ไปใช้ในการอ้างอิงสำหรับกำหนดระยะจริงจากตัวโปรแกรมควบคุม

ระยะที่ได้จากการทดลอง คือ $X = \frac{\text{ระยะจากการทดลอง}}{\text{จำนวน STEP ที่ใช้}}$, $X = \frac{1.6 \text{ cm}}{4000} = 4 \mu\text{m}$

X คือ ระยะของการเคลื่อนที่ต่อ 1 STEP

ระยะจากการคำนวณ คือ $x = \frac{h}{\text{จำนวน STEP ต่อ 1 รอบ}}$, $x = \frac{1.5 \text{ mm}}{400} = 3.75 \mu\text{m}$



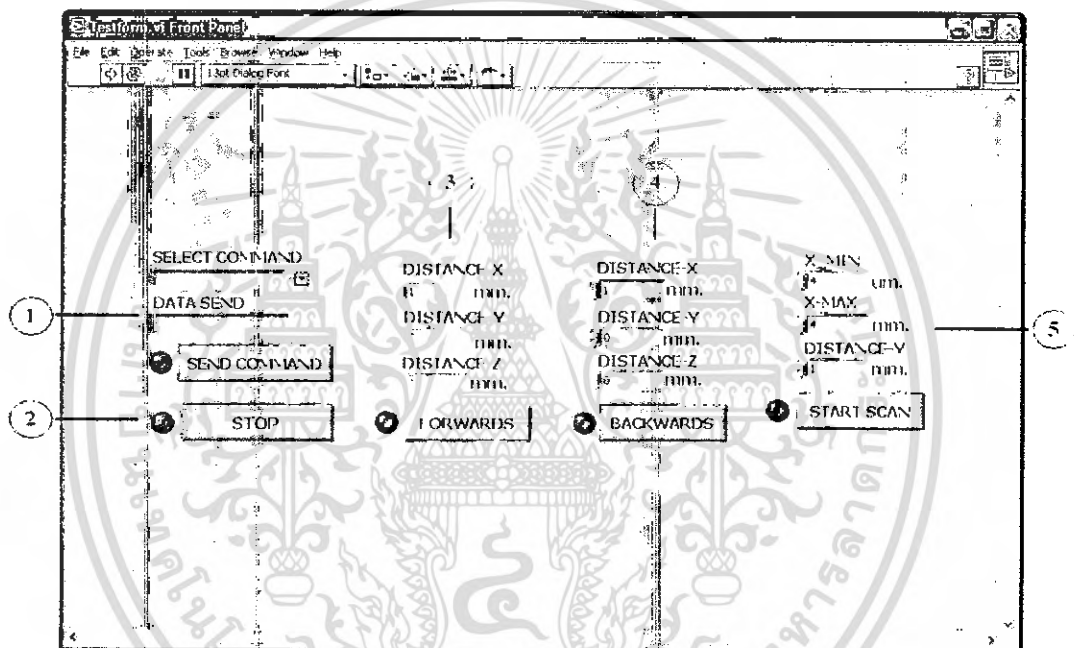
รูปที่ 7.7 แสดงการใช้เวอร์เนียร์วัดระยะการเคลื่อนที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7.3 การทดลองในส่วนของโปรแกรม LabVIEW

ในส่วนของโปรแกรม LabVIEW จะมีลักษณะของการทำงาน คือ เมื่อมีการสั่งงาน จากตัวโปรแกรมให้มีการทำงานในลักษณะต่าง ๆ ตัวโปรแกรมจะทำการส่งค่าตัวแปรออกไปยัง พอร์ตอนุกรมให้กับไมโครคอนโทรลเลอร์ เพื่อให้ไมโครคอนโทรลเลอร์ทำงานตามการสั่งงานจาก ตัวโปรแกรม LabVIEW

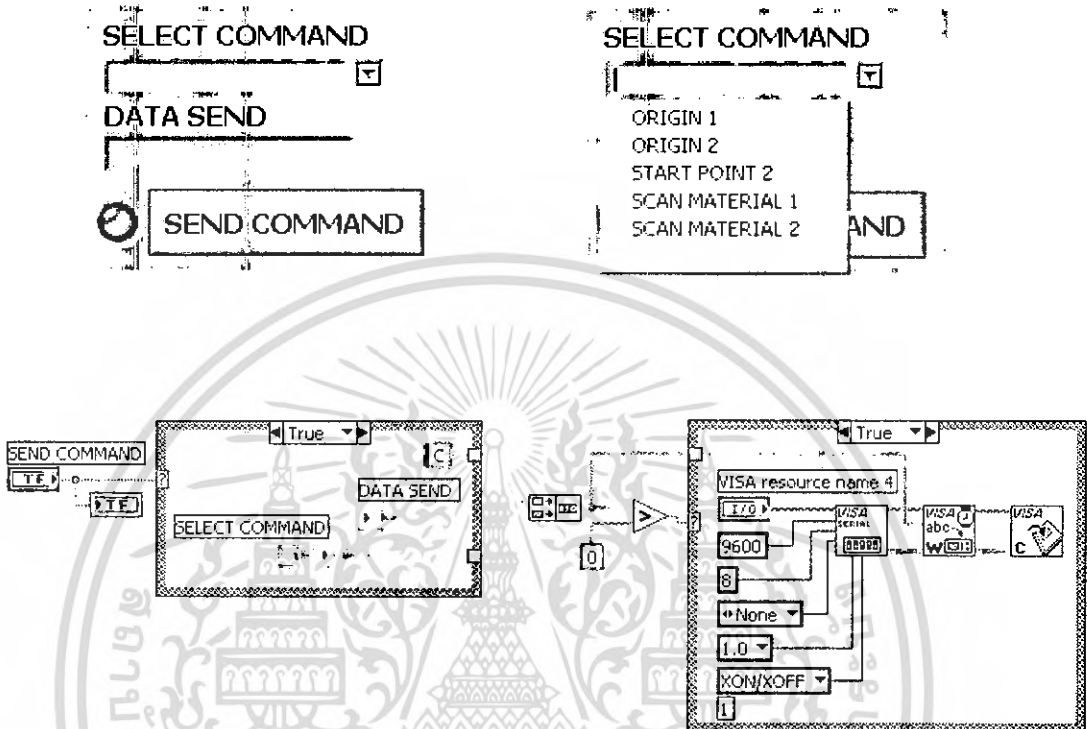
การทดลองจะแยกออกเป็น 5 ส่วน โดยมีรายละเอียดของส่วนต่างๆ ดังต่อไปนี้ คือ



รูปที่ 7.8 แสดงส่วนของ Front Panel ของโปรแกรม LabVIEW ที่ใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หมายเลข 1 การควบคุมการทำงานในรูปแบบต่างๆ (Select Command)



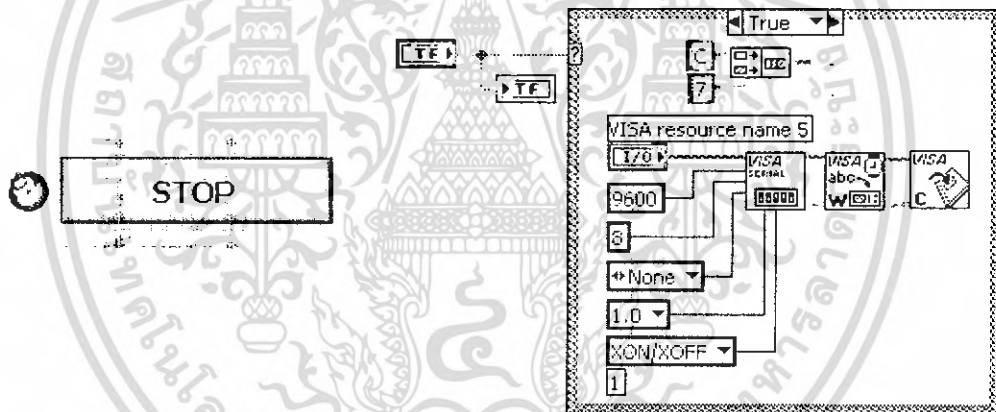
รูปที่ 7.9 แสดงส่วนของ Front Panel และ Block Diagram
ในส่วนของการควบคุมการทำงานในรูปแบบต่างๆ (Select Command)

เมื่อทำการเลือกรูปแบบการทำงานในช่อง Select Command แล้วทำการคลิกที่ปุ่ม Send Command ให้มีการเริ่มการทำงาน ตัวโปรแกรม LabVIEW จะทำการส่งค่าตัวแปรที่เป็นตัวแปรอักษร 2 ตัว โดยที่ตัวแปรตัวแรกจะเป็นตัวแปรที่บอกถึงการทำงานในส่วนของฟังก์ชันนี้หรือในส่วนของการ Select Command ตัวแปรตัวที่สองจะเป็นตัวแปรที่บอกถึงคำสั่งแต่ละคำสั่งที่ได้กำหนดขึ้น ค่าตัวแปรทั้งสองจะถูกส่งออกไปยังพอร์ตอนุกรมให้กับตัวไมโครคอนโทรลเลอร์ เพื่อให้ไมโครคอนโทรลเลอร์ทำงานตามรูปแบบของการทำงานที่ได้เลือกไว้

ในส่วนของการควบคุมการทำงานในรูปแบบต่างๆ นี้ได้กำหนดรูปแบบการทำงานไว้ 5 รูปแบบ ซึ่งแต่ละรูปแบบจะมีรายละเอียดของค่าตัวแปรดังนี้ คือ

คำสั่ง	ค่าตัวแปรที่หนึ่ง (ตัวแปรอักษร)	ค่าตัวแปรที่สอง (ตัวแปรอักษร)
Origin1	C	1
Origin2	C	2
Start Point2	C	3
Scan Material1	C	4
Scan Material2	C	5

หมายเลข 2 การสั่งหยุดการทำงาน (Stop)



รูปที่ 7.10 แสดงส่วนของ Front Panel และ Block Diagram

ในส่วนของการหยุดการทำงาน (Stop)

เมื่อมีการสั่งให้มีการหยุดการทำงานโดยการคลิกที่ปุ่ม Stop ตัวโปรแกรม LabVIEW จะทำการส่งค่าตัวแปรออกไปยังพอร์ตอนุกรมให้กับไมโครคอนโทรลเลอร์ทำงานในส่วนของการหยุดการทำงาน (Stop)

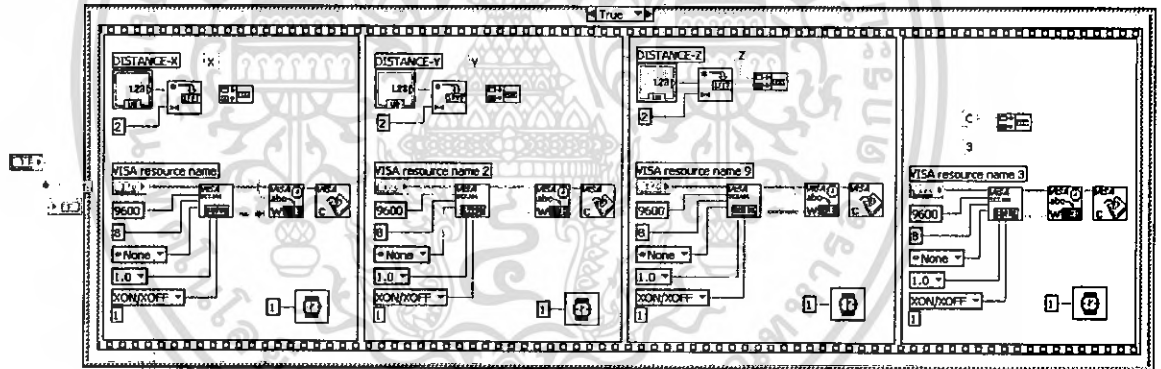
โดยในส่วนของคำสั่ง Stop ค่าตัวแปรที่โปรแกรม LabVIEW จะทำการส่งออกไปยังพอร์ตอนุกรมให้กับไมโครคอนโทรลเลอร์จะเป็นตัวแปรอักษร 2 ตัว โดยจะมีค่าเท่ากับ C7

หมายเลข 3 การสั่งงานให้มีการเคลื่อนที่ไปด้านหน้า (Forward)

DISTANCE-X
0 mm.
DISTANCE-Y
0 mm.
DISTANCE-Z
0 mm.

FORWARDS

DISTANCE-X = ช่องป้อนค่าระยะการเคลื่อนที่ไปด้านหน้าของแกน X
DISTANCE-Y = ช่องป้อนค่าระยะการเคลื่อนที่ไปด้านหน้าของแกน Y
DISTANCE-Z = ช่องป้อนค่าระยะการเคลื่อนที่ลงของแกน Z



รูปที่ 7.11 แสดงส่วนของ Front Panel และ Block Diagram
ในส่วนของการเคลื่อนที่ไปด้านหน้า (Forward)

เมื่อทำการสั่งงานจากตัวโปรแกรม LabVIEW ให้มีการทำงานในส่วนของการเคลื่อนที่ไปด้านหน้า โดยทำการป้อนค่าระยะทางที่ต้องการให้กับแกนแต่ละแกนในช่อง Distance ของแต่ละแกน (DISTANCE-X, DISTANCE-Y และ DISTANCE-Z) แล้วทำการคลิกที่ปุ่ม Forwards ให้มีการเริ่มการทำงาน ตัวโปรแกรม LabVIEW จะทำการส่งค่าตัวแปรออกไปยังพอร์ตอนุกรมให้กับไมโครคอนโทรลเลอร์ทำงานในส่วนของการเคลื่อนที่ไปด้านหน้า

โดยที่ค่าตัวแปรที่โปรแกรม LabVIEW จะทำการส่งออกไปยังพอร์ตอนุกรมเพื่อให้ไมโครคอนโทรลเลอร์ทราบถึงการทำงานในส่วนของฟังก์ชันนี้ หรือการทำงานในส่วนของการตั้งค่า Forward คือ ตัวแปรที่เป็นตัวอักษรซึ่งจะมีค่าเท่ากับ C3

ในส่วนของตัวแปรที่ใช้ในการส่งค่าของแกนแต่ละแกนจะมีรายละเอียดดังนี้ คือ

แกน	ตัวแปรบอกแกน (ตัวแปรอักษร)	ตัวแปรบอกระยะ (ตัวแปรตัวเลข)
X	X	0 - 255
Y	Y	0 - 255
Z	Z	0 - 255

หมายเลข 4 การสั่งงานให้มีการเคลื่อนที่ย้อนกลับ (Backward)

DISTANCE-X
[0] mm.

DISTANCE-Y
[0] mm.

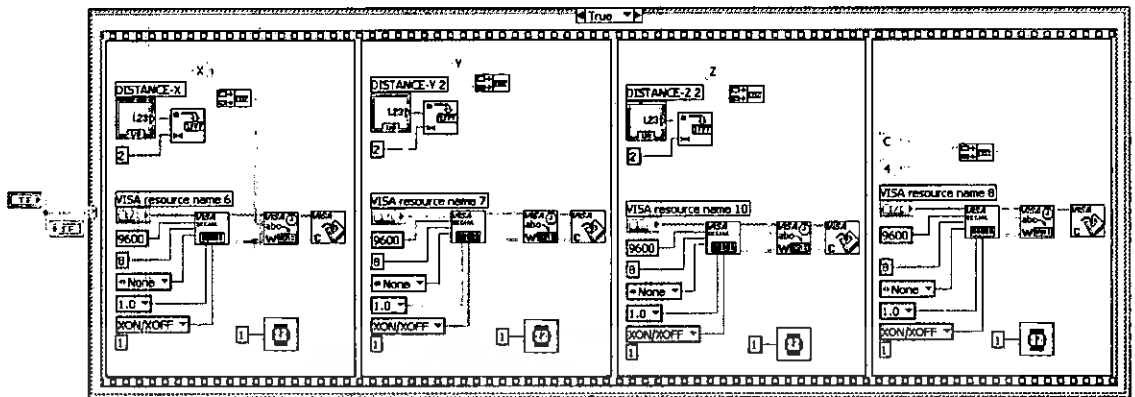
DISTANCE-Z
[0] mm.

BACKWARDS

DISTANCE-X = ช่องป้อนค่าระยะการเคลื่อนที่ย้อนกลับของแกน X

DISTANCE-Y = ช่องป้อนค่าระยะการเคลื่อนที่ย้อนกลับของแกน Y

DISTANCE-Z = ช่องป้อนค่าระยะการเคลื่อนที่ขึ้นของแกน Z



รูปที่ 7.12 แสดงส่วนของ Front Panel และ Block Diagram
ในส่วนของการเคลื่อนที่ย้อนกลับ (Backward)

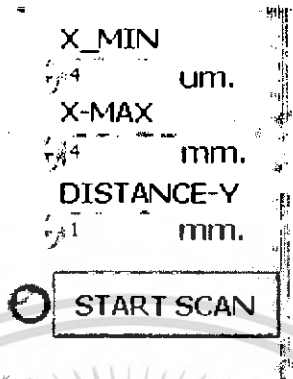
การทำงานจะเป็นไปในลักษณะเช่นเดียวกับการทำงานในส่วนของการเคลื่อนที่ไปด้านหน้า ซึ่งเมื่อทำการสั่งงานจากตัวโปรแกรม LabVIEW ให้มีการทำงานในส่วนของการเคลื่อนที่ย้อนกลับ โดยทำการป้อนค่าระยะทางที่ต้องการให้กับแกนแต่ละแกนในช่อง Distance ของแต่ละแกน (DISTANCE-X, DISTANCE-Y และ DISTANCE-Z) แล้วทำการคลิกที่ปุ่ม Backwards ให้มีการเริ่มการทำงาน ตัวโปรแกรม LabVIEW จะทำการส่งค่าตัวแปรออกไปยังพอร์ตอนุกรมให้กับไมโครคอนโทรลเลอร์ทำงานในส่วนของการเคลื่อนที่ย้อนกลับ

โดยที่ค่าตัวแปรที่โปรแกรม LabVIEW จะทำการส่งออกไปยังพอร์ตอนุกรมเพื่อให้ไมโครคอนโทรลเลอร์ทราบถึงการทำงานในส่วนของฟังก์ชันนี้ หรือการทำงานในส่วนของการสั่ง Backward คือ ตัวแปรที่เป็นตัวอักษรซึ่งจะมีค่าเท่ากับ C4

ในส่วนของตัวแปรที่ใช้ในการส่งค่าของแกนแต่ละแกนจะมีรายละเอียดดังนี้ คือ

แกน	ตัวแปรบอกแกน (ตัวแปรอักษร)	ตัวแปรบอกระยะ (ตัวแปรตัวเลข)
X	X	0 - 255
Y	Y	0 - 255
Z	Z	0 - 255

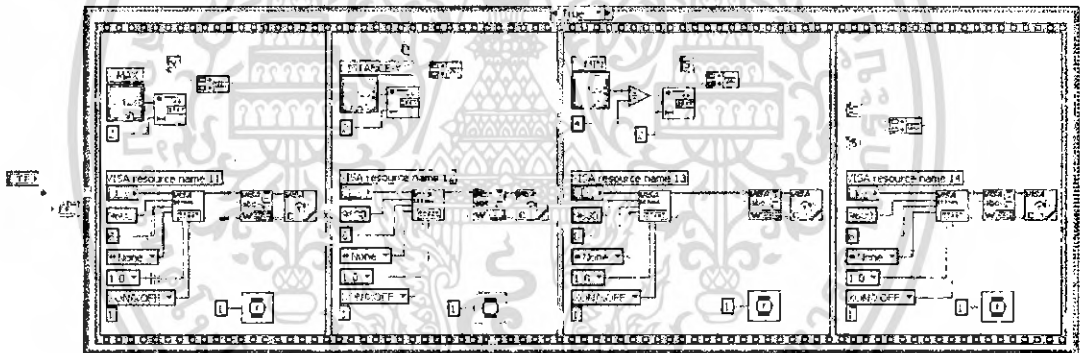
หมายเลข 5 การตั้งสแกน (Scan)



X_MIN = ช่องป้อนค่าระยะการสแกนของแกน X

X_MAX = ช่องป้อนค่าระยะการเคลื่อนที่ทั้งหมดของแกน X

DISTANCE-Y = ช่องป้อนค่าระยะการสแกนของแกน Y



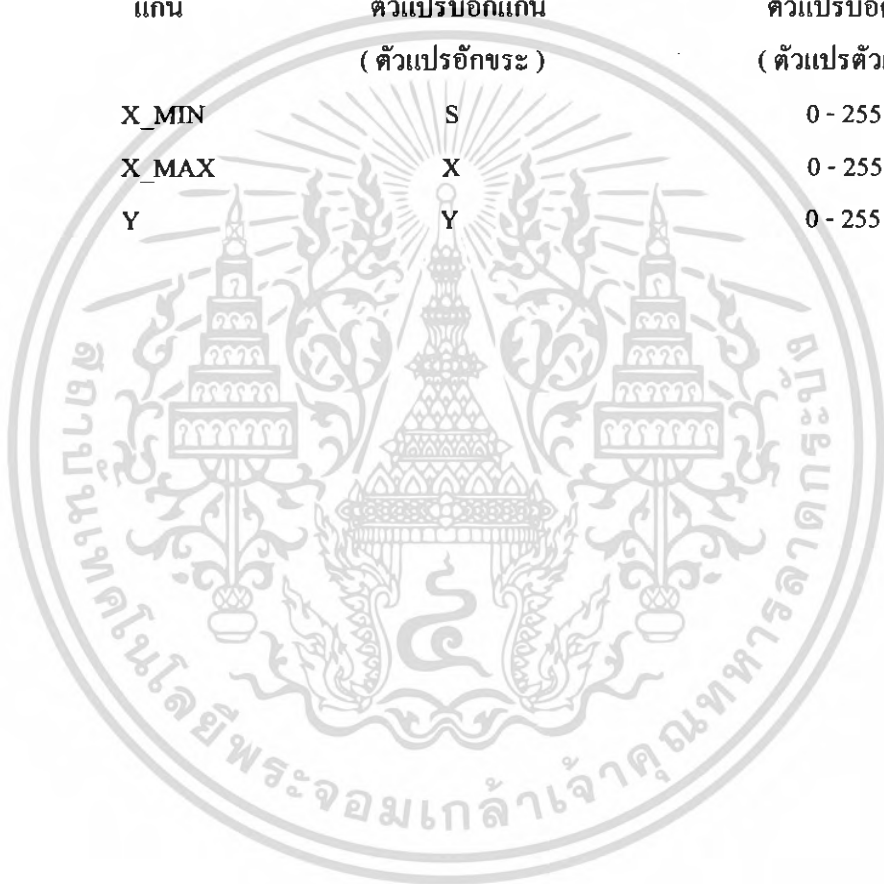
รูปที่ 7.13 แสดงส่วนของ Front Panel และ Block Diagram
ในส่วนของการสแกน (Scan)

เมื่อทำการสั่งงานจากตัวโปรแกรม LabVIEW ให้มีการทำงานในส่วนของการสแกน โดยทำการป้อนค่าระยะทางที่ต้องการในช่องของแกนแต่ละแกน (X_MIN, X_MAX และ DISTANCE-Y) แล้วทำการคลิกที่ปุ่ม Start Scan ให้มีการเริ่มการทำงาน ตัวโปรแกรม LabVIEW จะทำการส่งค่าตัวแปรออกไปยังพอร์ตอนุกรมให้กับไมโครคอนโทรลเลอร์ทำงานในส่วนของการสแกน

โดยค่าตัวแปรที่โปรแกรม LabVIEW จะทำการส่งออกไปให้กับไมโครคอนโทรลเลอร์ เพื่อให้ไมโครคอนโทรลเลอร์ทราบถึงการทำงานในส่วนของฟังก์ชันนี้ หรือการทำงานในส่วนของคำสั่ง Scan คือ ตัวแปรที่เป็นตัวอักษรซึ่งจะมีค่าเท่ากับ C5

ในส่วนของตัวแปรที่ใช้ในการส่งค่าของแกนแต่ละแกนจะรายละเอียดดังนี้ คือ

แกน	ตัวแปรบอกแกน (ตัวแปรอักษร)	ตัวแปรบอกระยะ (ตัวแปรตัวเลข)
X_MIN	S	0 - 255
X_MAX	X	0 - 255
Y	Y	0 - 255



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7.4 Source code สำหรับไมโครคอนโทรลเลอร์

```

#include <16F877.h>
#define TxD PIN_C6
#define RxD PIN_C7
#define CLOCK_SP 2000000
#fuses NOLVP,NOWDT,HS,NOPROTECT
#use delay (clock=CLOCK_SP)
#use rs232(baud=9600,xmit=TxD,rcv=RxD)
#include <stdlib.h>
#include <input.c>

//MOTOR1
#define ENA_X PIN_D0
#define CLK_X PIN_D1
#define DIR_X PIN_D2
//MOTOR1
#define ENA_Y PIN_D3
#define CLK_Y PIN_D4
#define DIR_Y PIN_D5
//MOTOR1
#define ENA_Z PIN_D6
#define CLK_Z PIN_D7
#define DIR_Z PIN_C4
//Limit Switch
#define SW_X1 input(PIN_A0)
#define SW_X2 input(PIN_A1)
#define SW_Y1 input(PIN_A2)
#define SW_Y2 input(PIN_A3)
#define SW_Z1 input(PIN_A4)
#define SW_Z2 input(PIN_A5)

#use fast_io(A)
#use fast_io(B)
#use fast_io(C)
#use fast_io(D)

int1
SW_LX1=1,SW_LX2=1,SW_LY1=1,SW_LY2=1,SW_LZ1
=1,SW_LZ2=1;
int1 ORG1,ORG2,STO,STP1,STP2,SCN;
int16
i,i1,i2,j,k,ST_X,ST_Y,ST_Z,X_MIN,SC_X,SC_Y,SC_Z,
SC_XMIN;
char DATA_COM,PORT_COM;
int16
DT_X=0,DT_Y=0,DT_Z=0,DT_XMIN=0,MOV_X,MOV
_Y,BUF_MOV;

//Function Phototype//
void origin_1(void);
void stop(void);
void scan(int16 SC_X , int16 SC_Y , int16 SC_XMIN);
void move_forward(int16 DT_X , int16 DT_Y , int16
DT_Z);
void move_backward(int16 DT_X , int16 DT_Y , int16
DT_Z);

//RS232 Interrupt Function
#INT_RDA

```

```

void rs232_isr()
{
    PORT_COM = getc();
    if(PORT_COM == 'X')
        DT_X = gethex();
    if(PORT_COM == 'Y')
        DT_Y = gethex();
    if(PORT_COM == 'Z')
        DT_Z = gethex();
    if(PORT_COM == 'S')
        DT_XMIN = gethex();
    if(PORT_COM == 'C')
        DATA_COM = getc();
    if(DATA_COM == '7')
    {
        STO = 1;
    } else STO = 0;
    disable_interrupts(INT_RDA);
    enable_interrupts(INT_RDA);
}

//Interrupt form Limit Switch Function
#INT_EXT
void interrupt_port_a(void)
{
    //Limit Switch AXIS-X
    if(SW_X1==0)
    {
        SW_LX1 = 0;
        delay_ms(20);
    }
    else SW_LX1 = 1;

    if(SW_X2==0)
    {
        SW_LX2 = 0;
        delay_ms(20);
    }
    else SW_LX2 = 1;

    //Limit Switch AXIS-Y
    if(SW_Y1==0)
    {
        SW_LY1 = 0;
        delay_ms(20);
    }
    else SW_LY1 = 1;
    if(SW_Y2==0)
    {
        SW_LY2 = 0;
        delay_ms(20);
    }
    else SW_LY2 = 1;

    //Limit Switch AXIS-Z
    if(SW_Z1==0)
    {
        SW_LZ1 = 0;
        delay_ms(20);
    }
    else SW_LZ1 = 1;
    if(SW_Z2==0)

```

```

{
    SW_LZ2 = 0;
    delay_ms(20);
}
else SW_LZ2 = 1;
disable_interrupts(INT_EXT);
enable_interrupts(INT_EXT);
}

//Function origin//
void origin_1(void)
{
    output_high(ENA_X);
    output_high(ENA_Y);
    output_high(ENA_Z);
    SW_LX1 = 1;
    SW_LY1 = 1;
    SW_LZ1 = 1;
    While(SW_LZ1)
    {
        output_high(DIR_Z);
        output_high(CLK_Z);
        delay_us(1200);
        output_low(CLK_Z);
        delay_us(1200);
        if(STO == 1)
        {
            stop();
            break;
        }
    }
}
SW_LZ1 = 1;

for(i=0; i<500; i++)
{
    output_low(DIR_Z);
    output_high(CLK_Z);
    delay_us(1100);
    output_low(CLK_Z);
    delay_us(1100);
    if(STO == 1)
    {
        stop();
        break;
    }
}
While(SW_LX1 && SW_LY1)
{
    output_low(DIR_X);
    output_low(DIR_Y);
    output_high(CLK_X);
    output_high(CLK_Y);
    delay_us(1050);
    output_low(CLK_X);
    output_low(CLK_Y);
    delay_us(1050);
    if(STO == 1)
    {
        stop();
        break;
    }
}
if(SW_LX1 == 0)
{
    for(i=0; i<1250; i++)

```

```

{
    output_high(DIR_X);
    output_high(CLK_X);
    delay_us(1000);
    output_low(CLK_X);
    delay_us(1000);
    if(STO == 1)
    {
        stop();
        break;
    }
}
SW_LX1 = 1;
While(SW_LY1)
{
    output_low(DIR_Y);
    output_high(CLK_Y);
    delay_us(950);
    output_low(CLK_Y);
    delay_us(950);
    if(STO == 1)
    {
        stop();
        break;
    }
}
SW_LY1 = 1;
for(i=0; i<2500; i++)
{
    output_high(DIR_Y);
    output_high(CLK_Y);
    delay_us(950);
    output_low(CLK_Y);
    delay_us(950);
    if(STO == 1)
    {
        stop();
        break;
    }
}
SW_LY1 = 1;
While(SW_LX1)
{
    output_low(DIR_X);
    output_high(CLK_X);
    delay_us(1000);
    output_low(CLK_X);
    delay_us(1000);
}
}
    output_low(CLK_Y);
    delay_us(950);
    if(STO == 1)
    {
        stop();
        break;
    }
}
    if(SW_LY1 == 0)
    {
        for(i=0; i<2500; i++)
        {
            output_high(DIR_Y);
            output_high(CLK_Y);
            delay_us(950);
            output_low(CLK_Y);
            delay_us(950);
            if(STO == 1)
            {
                stop();
                break;
            }
        }
        SW_LY1 = 1;
    }
}
    While(SW_LX1)
    {
        output_low(DIR_X);
        output_high(CLK_X);
        delay_us(1000);
        output_low(CLK_X);
        delay_us(1000);
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if(STO == 1)
{
    stop();
    break;
}
}
SW_LX1 = 1;
for(i=0; i<1250; i++)
{
    output_high(DIR_X);
    output_high(CLK_X);
    delay_us(1000);
    output_low(CLK_X);
    delay_us(1000);
    if(STO == 1)
    {
        stop();
        break;
    }
}
stop();
break;

void origin_2(void)
{
    SW_LX1 = 1;
    SW_LY1 = 1;
    output_high(ENA_X);
    output_high(ENA_Y);
    output_low(ENA_Z);
    while(SW_LX1)

```

```

{
    output_low(DIR_X);
    output_high(CLK_X);
    delay_us(1000);
    output_low(CLK_X);
    delay_us(1000);
    if(STO == 1)
    {
        stop();
        break;
    }
}
for(i=0; i<1250; i++)
{
    output_high(DIR_X);
    output_high(CLK_X);
    delay_us(1000);
    output_low(CLK_X);
    delay_us(1000);
    if(STO == 1)
    {
        stop();
        break;
    }
}
while(SW_LY1)
{
    output_low(DIR_Y);
    output_high(CLK_Y);
    delay_us(950);
    output_low(CLK_Y);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

delay_us(950);
if(STO == 1)
{
    stop();
    break;
}
for(i=0; i<2500; i++)
{
    output_high(DIR_Y);
    output_high(CLK_Y);
    delay_us(950);
    output_low(CLK_Y);
    delay_us(950);
    if(STO == 1)
    {
        stop();
        break;
    }
}
stop();
break;
}

//Function move_forward//
void move_forward(int16 DT_X , int16 DT_Y ,
int16 DT_Z)
{
    output_high(ENA_X);
    output_high(ENA_Y);
    output_high(ENA_Z);
    ST_X = DT_X*250;
    ST_Y = DT_Y*250;
    ST_Z = DT_Z*250;
    if(ST_X > ST_Y)
    {
        BUF_MOV = ST_X - ST_Y;
        SW_LX2 = 1;
        SW_LY2 = 1;
        i = 0;
        while(SW_LX2 && SW_LY2 && i<ST_Y)
        {
            output_high(DIR_X);
            output_high(DIR_Y);
            output_high(CLK_X);
            output_high(CLK_Y);
            delay_us(1100);
            output_low(CLK_X);
            output_low(CLK_Y);
            delay_us(1100);
            i++;
            if(STO == 1)
            {
                stop();
                break;
            }
        }
    }
    if(i==ST_Y)
    {
        for(j=0; j<BUF_MOV; j++)
        {
            output_high(DIR_X);
            output_high(CLK_X);

```



```

delay_us(1000);
output_low(CLK_Y);
delay_us(1000);
if(STO == 1)
{
    stop();
    break;
}
}
SW_LY2 = 1;
}
}
if(SW_LY2==0)
{
    for(j=0; j<1250; j++)
    {
        output_low(DIR_Y);
        output_high(CLK_Y);
        delay_us(1000);
        output_low(CLK_Y);
        delay_us(1000);
        if(STO == 1)
        {
            stop();
            break;
        }
    }
}
SW_LY2 = 1;
while(SW_LX2 && i<ST_Y)
{
    output_high(DIR_X);
    output_high(CLK_X);
    delay_us(1000);
    output_low(CLK_X);
    delay_us(1000);
    if(STO == 1)
    {
        stop();
        break;
    }
    i++;
}
if(SW_LX2==0)
{
    for(j=0; j<1250; j++)
    {
        output_low(DIR_X);
        output_high(CLK_X);
        delay_us(1000);
        output_low(CLK_X);
        delay_us(1000);
        if(STO == 1)
        {
            stop();
            break;
        }
    }
    SW_LX2 = 1;
}
else
{
    for(j=0; j<BUF_MOV; j++)
    {
        output_high(DIR_X);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

output_high(CLK_X);
delay_us(1000);
output_low(CLK_X);
delay_us(1000);
if(STO == 1)
{
    stop();
    break;
}
if(SW_LX2==0)
{
    for(j=0; j<1250; j++)
    {
        output_low(DIR_X);
        output_high(CLK_X);
        delay_us(1000);
        output_low(CLK_X);
        delay_us(1000);
        if(STO == 1)
        {
            stop();
            break;
        }
    }
    SW_LX2 = 1;
}
}
}
}

if(ST_Y > ST_X)
{
    BUF_MOV = ST_Y - ST_X;
    SW_LX2 = 1;
    SW_LY2 = 1;
    i = 0;
    while(SW_LX2 && SW_LY2 && i<ST_X)
    {
        output_high(DIR_X);
        output_high(DIR_Y);
        output_high(CLK_X);
        output_high(CLK_Y);
        delay_us(1100);
        output_low(CLK_X);
        output_low(CLK_Y);
        delay_us(1100);
        i++;
        if(STO == 1)
        {
            stop();
            break;
        }
    }
    if(i==ST_X)
    {
        for(j=0; j<BUF_MOV; j++)
        {
            output_high(DIR_Y);
            output_high(CLK_Y);
            delay_us(1000);
            output_low(CLK_Y);
            delay_us(1000);
            if(STO == 1)

```

```

{
    stop();
    break;
}
if(SW_LY2==0)
{
    for(j=0; j<1250; j++)
    {
        output_low(DIR_Y);
        output_high(CLK_Y);
        delay_us(1000);
        output_low(CLK_Y);
        delay_us(1000);
        if(STO == 1)
        {
            stop();
            break;
        }
        SW_LY2 = 1;
    }
}
i = 0;
}
if(SW_LY2==0)
{
    for(j=0; j<1250; j++)
    {
        output_low(DIR_Y);
        output_high(CLK_Y);
        delay_us(1000);
        output_low(CLK_Y);
        delay_us(1000);
        if(STO == 1)
        {
            stop();
            break;
        }
        SW_LY2 = 1;
    }
}
if(SW_LX2==0)
{
    for(j=0; j<1250; j++)
    {
        output_low(DIR_X);
        output_high(CLK_X);
        delay_us(1000);
        output_low(CLK_X);
        delay_us(1000);
        if(STO == 1)
        {
            stop();
            break;
        }
        SW_LX2 = 1;
    }
}
if(SW_LX2==0)
{
    for(j=0; j<1250; j++)
    {
        output_low(DIR_X);
        output_high(CLK_X);
        delay_us(1000);
        output_low(CLK_X);
        delay_us(1000);
        if(STO == 1)
        {
            stop();
            break;
        }
        SW_LX2 = 1;
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    {
        stop();
        break;
    }
}
SW_LX2 = 1;
}
}

if(SW_LX2==0)
{
    for(j=0; j<1250; j++)
    {
        output_low(DIR_X);
        output_high(CLK_X);
        delay_us(1000);
        output_low(CLK_X);
        delay_us(1000);
        if(STO == 1)
        {
            stop();
            break;
        }
    }
    SW_LX2 = 1;
    while(SW_LY2 && i<ST_X)
    {
        output_high(DIR_Y);
        output_high(CLK_Y);
        delay_us(1000);
        output_low(CLK_Y);
        delay_us(1000);
        i++;
        if(STO == 1)
        {
            stop();
            break;
        }
    }
    if(SW_LY2==0)
    {
        for(j=0; j<1250; j++)
        {
            output_low(DIR_Y);
            output_high(CLK_Y);
            delay_us(1000);
            output_low(CLK_Y);
            delay_us(1000);
            if(STO == 1)
            {
                stop();
                break;
            }
        }
        SW_LY2 = 1;
    }
    else
    {
        for(j=0; j<BUF_MOV; j++)
        {
            output_high(DIR_Y);
            output_high(CLK_Y);
            delay_us(1000);
            output_low(CLK_Y);
        }
    }
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

delay_us(1000);
if(STO == 1)
{
    stop();
    break;
}
if(SW_LY2==0)
{
    for(j=0; j<1250; j++)
    {
        output_low(DIR_Y);
        output_high(CLK_Y);
        delay_us(1000);
        output_low(CLK_Y);
        delay_us(1000);
        if(STO == 1)
        {
            stop();
            break;
        }
    }
    SW_LY2 = 1;
}
}
}

if(ST_X==ST_Y)
{
    SW_LX2 = 1;
    SW_LY2 = 1;

    i = 0;
    while(SW_LX2 && SW_LY2 && i<ST_Y)
    {
        output_high(DIR_X);
        output_high(DIR_Y);
        output_high(CLK_X);
        output_high(CLK_Y);
        delay_us(1100);
        output_low(CLK_X);
        output_low(CLK_Y);
        delay_us(1100);
        i++;
        if(STO == 1)
        {
            stop();
            break;
        }
        if(SW_LX2==0)
        {
            for(j=0; j<1250; j++)
            {
                output_low(DIR_X);
                output_high(CLK_X);
                delay_us(1000);
                output_low(CLK_X);
                delay_us(1000);
                if(STO == 1)
                {
                    stop();
                    break;
                }
            }
        }
    }
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


```

        if(STO == 1)
        {
            stop();
            break;
        }
        SW_LX1 = 1;
    }
}
i = 0;
}
}
if(SW_LX1==0)
{
    for(j=0; j<1250; j++)
    {
        output_high(DIR_X);
        output_high(CLK_X);
        delay_us(1000);
        output_low(CLK_X);
        delay_us(1000);
        if(STO == 1)
        {
            stop();
            break;
        }
    }
    SW_LX1 = 1;
    while(SW_LY1 && i<ST_Y)
    {
        output_low(DIR_Y);
        output_high(CLK_Y);
        delay_us(1000);
        output_low(CLK_Y);
        delay_us(1000);
        if(STO == 1)
        {
            stop();
            break;
        }
        i++;
        if(STO == 1)
        {
            stop();
            break;
        }
    }
    SW_LY1 = 0;
    for(j=0; j<1250; j++)
    {
        output_high(DIR_Y);
        output_high(CLK_Y);
        delay_us(1000);
        output_low(CLK_Y);
        delay_us(1000);
        if(STO == 1)
        {
            stop();
            break;
        }
    }
    SW_LY1 = 1;
}
}
if(SW_LY1==0)
{
    for(j=0; j<1250; j++)
    {
        output_high(DIR_Y);
        output_high(CLK_Y);
        delay_us(1000);
        output_low(CLK_Y);
        delay_us(1000);
        if(STO == 1)
        {
            stop();
            break;
        }
    }
    SW_LY1 = 1;
}
}
if(SW_LY1==0)
{
    for(j=0; j<1250; j++)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{
    output_high(DIR_Y);
    output_high(CLK_Y);
    delay_us(1000);
    output_low(CLK_Y);
    delay_us(1000);
    if(STO == 1)
    {
        stop();
        break;
    }
}
SW_LY1 = 1;
while(SW_LX1 && i < ST_Y)
{
    output_low(DIR_X);
    output_high(CLK_X);
    delay_us(1000);
    output_low(CLK_X);
    delay_us(1000);
    i++;
    if(STO == 1)
    {
        stop();
        break;
    }
}
if(SW_LX1 == 0)
{
    for(j=0; j < 1250; j++)
    {
        output_high(DIR_X);
        output_high(CLK_X);
        delay_us(1000);
        output_low(CLK_X);
        delay_us(1000);
        if(STO == 1)
        {
            stop();
            break;
        }
    }
}
if(SW_LX1 == 0)
{
    for(j=0; j < 1250; j++)
    {
        output_high(DIR_X);
        output_high(CLK_X);
        delay_us(1000);
        output_low(CLK_X);
        delay_us(1000);
        if(STO == 1)
        {
            stop();
            break;
        }
    }
}
SW_LX1 = 1;
else
{
    for(j=0; j < BUF_MOV; j++)
    {
        output_low(DIR_X);
        output_high(CLK_X);
        delay_us(1000);
        output_low(CLK_X);
        delay_us(1000);
        if(STO == 1)
        {
            stop();
            break;
        }
    }
}
if(SW_LX1 == 0)
{
    for(j=0; j < 1250; j++)
    {
        output_high(DIR_X);
        output_high(CLK_X);
        delay_us(1000);
        output_low(CLK_X);
        delay_us(1000);
        if(STO == 1)
        {
            stop();
            break;
        }
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

delay_us(1000);
output_low(CLK_X);
delay_us(1000);
if(STO == 1)
{
    stop();
    break;
}
}
}
SW_LX1 = 1;
}
}
}
}
if(ST_Y > ST_X)
{
    BUF_MOV = ST_Y - ST_X;
    SW_LX1 = 1;
    SW_LY1 = 1;
    i = 0;
    while(SW_LX1 && SW_LY1 &&
i<ST_X)
    {
        output_low(DIR_X);
        output_low(DIR_Y);
        output_high(CLK_X);
        output_high(CLK_Y);
        delay_us(1100);
        output_low(CLK_X);
        output_low(CLK_Y);
        delay_us(1100);
        i++;
        if(STO == 1)
        {
            stop();
            break;
        }
    }
    if(i==ST_X)
    {
        for(j=0; j<BUF_MOV; j++)
        {
            output_low(DIR_Y);
            output_high(CLK_Y);
            delay_us(1000);
            output_low(CLK_Y);
            delay_us(1000);
            if(STO == 1)
            {
                stop();
                break;
            }
        }
        if(SW_LY1==0)
        {
            for(j=0; j<1250; j++)
            {
                output_high(DIR_Y);
                output_high(CLK_Y);
                delay_us(1000);
                output_low(CLK_Y);
                delay_us(1000);
                if(STO == 1)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


```

output_high(CLK_X);
delay_us(1000);
output_low(CLK_X);
delay_us(1000);
if(STO == 1)
{
    stop();
    break;
}
}
SW_LX1 = 1;
while(SW_LY2 && i < ST_X)
{
    output_low(DIR_Y);
    output_high(CLK_Y);
    delay_us(1000);
    output_low(CLK_Y);
    delay_us(1000);
    i++;
    if(STO == 1)
    {
        stop();
        break;
    }
}
if(SW_LY1 == 0)
{
    for(j=0; j<1250; j++)
    {
        output_high(DIR_Y);
        output_high(CLK_Y);
        delay_us(1000);
        output_low(CLK_Y);
        output_low(CLK_Y);
        delay_us(1000);
        if(STO == 1)
        {
            stop();
            break;
        }
    }
}
else
{
    for(j=0; j<BUF_MOV; j++)
    {
        output_low(DIR_Y);
        output_high(CLK_Y);
        delay_us(1000);
        output_low(CLK_Y);
        delay_us(1000);
        if(STO == 1)
        {
            stop();
            break;
        }
    }
    if(SW_LY1 == 0)
    {
        for(j=0; j<1250; j++)
        {
            output_high(DIR_Y);
            output_high(CLK_Y);
            delay_us(1000);
            output_low(CLK_Y);
        }
    }
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


```

        break;
    }
}
if(SW_LY1==0)
{
    for(j=0; j<1250; j++)
    {
        output_high(DIR_Y);
        output_high(CLK_Y);
        delay_us(1000);
        output_low(CLK_Y);
        delay_us(1000);
        if(STO == 1)
        {
            stop();
            break;
        }
    }
    SW_LY1 = 1;
}
}

if(SW_LY1==0)
{
    for(j=0; j<1250; j++)
    {
        output_high(DIR_Y);
        output_high(CLK_Y);
        delay_us(1000);
        output_low(CLK_Y);
        delay_us(1000);
        if(STO == 1)
        {
            stop();
            break;
        }
    }
}

if(SW_LY1==0)
{
    for(j=0; j<1250; j++)
    {
        output_high(DIR_X);
        output_high(CLK_X);
        delay_us(1000);
        output_low(CLK_X);
        delay_us(1000);
        if(STO == 1)
        {
            stop();
            break;
        }
    }
}

if(SW_LX1==0)
{
    for(j=0; j<1250; j++)
    {
        output_low(DIR_X);
        output_high(CLK_X);
        delay_us(1000);
        output_low(CLK_X);
        delay_us(1000);
        if(STO == 1)
        {
            stop();
            break;
        }
    }
}
if(SW_LX1==0)
{
    for(j=0; j<1250; j++)
    {
        output_high(DIR_X);
        output_high(CLK_X);
        delay_us(1000);
        output_low(CLK_X);
        delay_us(1000);
        if(STO == 1)
        {
            stop();
            break;
        }
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


```

for(i=0; i<DT_XMIN; i++)
{
    output_low(DIR_X);
    output_high(CLK_X);
    delay_us(1200);
    output_low(CLK_X);
    delay_us(1200);
    if(STO == 1)
    {
        stop();
        break;
    }
}

for(i=0; i<=ST_Y; i++)
{
    output_low(DIR_Y);
    output_high(CLK_Y);
    delay_us(1000);
    output_low(CLK_Y);
    delay_us(1000);
    if(STO == 1)
    {
        stop();
        break;
    }
}

//Function stop//
void stop(void)
{
    output_low(ENA_X);
    output_low(ENA_Y);
    output_low(ENA_Z);
}

//Function main//
void main(void)
{
    set_tris_a(0xFF);
    set_tris_c(0x80);
    set_tris_d(0x00);

    //disable_interrupts(GLOBAL);
    enable_interrupts(GLOBAL);
    enable_interrupts(INT_RDA);
    enable_interrupts(INT_EXT);
    delay_ms(10);
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

DATA_COM = '0';          }
while(1)
{
  if(DATA_COM == '1')
  {
    origin_10;
    DATA_COM = '0';
  }
  if(DATA_COM == '2')
  {
    origin_20;
    DATA_COM = '0';
  }
  if(DATA_COM == '3')
  {
    move_forward(DT_X,DT_Y,DT_Z);
    DATA_COM = '0';
  }
  if(DATA_COM == '4')
  {
    move_backward(DT_X,DT_Y,DT_Z);
    DATA_COM = '0';
  }
  if(DATA_COM == '5')
  {
    scan(DT_X,DT_Y, DT_XMIN);
    DATA_COM = '0';
  }
  if(DATA_COM == '7')
  {
  }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปัญหาที่พบในการดำเนินงาน

1. ปัญหาในการสร้างระบบ Mechanics เนื่องจากไม่มีความรู้ทางด้านนี้ จึงทำให้การทำงานตรงจุดนี้ค่อนข้างที่จะล่าช้า
2. เนื่องจากใช้เกลิยวสกรูเป็นแบบเกลิยวสเดนเลส และมีขนาดเส้นผ่านศูนย์กลางเล็ก จึงทำให้เกิดการโค้งงอได้ง่าย ส่งผลกระทบทำให้มีปัญหาในการเคลื่อนที่ และการกำหนดระยะที่ถูกต้อง
3. ความเร็วในการเคลื่อนที่ของแต่ละแกนค่อนข้างช้า เป็นผลจากระบบแมคคานิค ที่ต้องมาปรับปรุงให้มีประสิทธิภาพมากขึ้น

สรุปผลการทดลอง

ส่วนของการทดลองวัดสัญญาณ รูปร่างของสัญญาณที่วัดได้จากออปโตมิเตอร์รูปร่างไม่แตกต่างจากที่ออกมาจากไมโครคอนโทรลเลอร์ เพราะเนื่องมาจากสัญญาณที่ใช้ป้อนให้กับไอซี L297 มีความถี่ไม่สูงมากนัก

ส่วนของการทดลองหาระยะการเคลื่อนที่ ใน 1 STEP จะเคลื่อนที่ได้ระยะเท่าไร ซึ่งจากการทดลองเราได้ผลของระยะการเคลื่อนที่ต่อ 1 STEP เท่ากับ $0.4 \mu\text{m}$ และระยะจากการคำนวณ $0.375 \mu\text{m}$ จะเห็นว่ามีความใกล้เคียงกัน แตกต่างกันเพียง $0.025 \mu\text{m}$ ค่าผิดพลาดดังกล่าวอาจเกิดได้จากหลายสาเหตุด้วยกัน อาทิ ค่าผิดพลาดจากระบบกลไกแมคคานิค เช่น ระยะพิทของเกลิยวไม่เท่ากันตลอดทั้งอัน, ช่องว่างที่เกิดการคราฟเกลิยว

ในส่วนของโปรแกรม LabVIEW แบ่งการสั่งงานเป็นฟังก์ชันหลักๆ ดังนี้คือ ฟังก์ชันจุดเริ่มต้น (ORIGIN), ฟังก์ชันเดินหน้า (FORWARD) และ ถอยหลัง (BACK WARD), ฟังก์ชันการสแกน (SCAN), ฟังก์ชันหยุดการทำงาน (STOP)

เมื่อมีการสั่งงานจากตัวโปรแกรม ตัวโปรแกรมจะทำการส่งค่าตัวแปรที่เป็นตัวแปรอักขระ 2 ตัวออกไปยังพอร์ตอนุกรมให้กับไมโครคอนโทรลเลอร์ เพื่อให้ไมโครคอนโทรลเลอร์ทำงานตามรูปแบบของการทำงานที่ได้เลือกไว้ โดยที่ตัวแปรตัวแรกจะเป็นตัวแปรที่บอกถึงการทำงานในส่วนของฟังก์ชันนี้ ตัวแปรตัวที่สองจะเป็นตัวแปรที่บอกถึงคำสั่งแต่ละคำสั่งที่ได้กำหนดขึ้น

บทที่ 8

บทสรุป

โครงการนี้ เป็นการสร้างเครื่องสแกนระบบ 3 มิติ ซึ่งจะใช้ในการสแกนหารอยร้าวในเนื้อโลหะ โดยการคิดหัวอัลตราโซนิกไว้กับตัวเครื่อง ตัวเครื่องจะประกอบด้วยแกนทั้งหมด 3 แกน คือ แกน X, แกน Y, และแกน Z โดยแต่ละแกนจะถูกขับเคลื่อนด้วยตัวสเต็ปปีงมอเตอร์ผ่านทางเกลิยวสกรู ในส่วนของการทำงานของสเต็ปปีงมอเตอร์จะถูกควบคุมการทำงานโดยไมโครคอนโทรลเลอร์อีกทีหนึ่ง และในส่วนของการทำงานทั้งหมดของตัวเครื่อง จะทำการควบคุมการทำงานผ่านทางโปรแกรม LabVIEW ซึ่งตัวโปรแกรม LabVIEW จะเป็นตัวที่ทำหน้าที่ในการส่งค่าของตัวแปรผ่านทางพอร์ตอนุกรม เพื่อติดต่อและควบคุมการทำงานของไมโครคอนโทรลเลอร์

โครงสร้างหลักของเครื่องจะแบ่งออกเป็น 2 ส่วน คือ

1. ส่วนของฮาร์ดแวร์
2. ส่วนของซอฟต์แวร์

1. ส่วนของฮาร์ดแวร์

ในส่วนของฮาร์ดแวร์ก็จะแบ่งออกเป็น 2 ส่วน คือ 1. ส่วนของชุดแมคคานิก ซึ่งก็คือส่วนของโครงสร้างทั้งหมดของตัวเครื่อง ทำขึ้นด้วยอลูมิเนียมซึ่งจะมีน้ำหนักเบาและมีความแข็งแรง ในการควบคุมการเคลื่อนที่ของแกนแต่ละแกนจะทำการควบคุมโดยสเต็ปปีงมอเตอร์ ส่งกำลังผ่านทางเกลิยวสกรูโดยมีสไลด์บาร์เป็นไกด์หรือเป็นตัวประกอบสำหรับการเคลื่อนที่ของแกนแต่ละแกน และ 2. ส่วนของวงจรที่ใช้ในการควบคุมการทำงานของสเต็ปปีงมอเตอร์ ซึ่งก็คือวงจรของตัวไมโครคอนโทรลเลอร์ และวงจรขับเคลื่อนสเต็ปปีงมอเตอร์ของแต่ละแกน

2. ส่วนของซอฟต์แวร์

ในส่วนของซอฟต์แวร์ก็จะแบ่งออกเป็น 2 ส่วน คือ 1. ส่วนของโปรแกรมภาษาซีที่ใช้ในการควบคุมการทำงานของไมโครคอนโทรลเลอร์ 2. ส่วนของโปรแกรม LabVIEW เป็นส่วนที่ใช้ติดต่อกับไมโครคอนโทรลเลอร์ผ่านทางพอร์ตอนุกรม

โครงการเครื่องสแกนระบบ 3 มิติที่สร้างขึ้นนี้ สามารถที่จะนำไปประยุกต์ใช้งานกับงานในด้านต่าง ๆ ได้ อาทิเช่น เครื่องแกะสลักไม้, เครื่องเจาะแผ่นปริ้น และอาจจะนำไปสแกนชิ้นงานอื่น ๆ ได้ เป็นต้น

คุณลักษณะของเครื่องสแกนระบบ 3 มิติ

- เครื่องสแกนระบบ 3 มิติ นี้มีความละเอียดในการเคลื่อนที่ $4 \mu\text{m}$ ต่อ 1 Step
- ระยะทางที่สามารถเคลื่อนที่ได้ แกน $x = 25 \text{ cm}$, แกน $y = 22 \text{ cm}$, แกน $z = 10 \text{ cm}$

แนวทางการพัฒนา

1. ออกแบบปรับปรุงระบบเมคานิกให้มีการเคลื่อนที่ราบรื่นมากขึ้น โดยการนำแกนเกลียวที่เป็น BallScrew มาใช้งานแทนเกลียวสแตนเลส
2. ปรับปรุงโปรแกรม LabVIEW และ Microcontroller ให้มีประสิทธิภาพมากขึ้น
3. เพิ่มรูปแบบการเคลื่อนที่ให้หลากหลายมากขึ้น
4. ปรับปรุงชุดไคร์ฟสตีปิ้งมอเตอร์ให้สามารถไคร์ฟที่ความเร็วสูงๆ ได้ดีขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

1. ประจัน พลังสันติกุล, เรียนรู้และใช้งาน CCS C คอมไพเลอร์เขียนโปรแกรมภาษา C ความคุม ไมโครคอนโทรลเลอร์ PIC (PIC Microcontroller programming with CCS C compiler), กรุงเทพฯ : อินโนเวตีฟ เอ็กเพอริเมนต์, 312 หน้า
2. เจริญ เพชรมณี, “เรียนลัด LabVIEW ”, กรุงเทพฯ : ซีเอ็ดยูเคชั่น , 216 หน้า, 2547
3. เศรษฐี มณีธรรม และ สำเริง เต็มราม, “ คำภีร์ไมโครคอนโทรลเลอร์ PIC (Microcontroller PIC) ”, กรุงเทพฯ : เคทีพี คอมพ์ แอนด์ คอนซัลท์, 368 หน้า , 2549
4. ญัฐพล วงศ์สุนทรชัย และ ชัยวัฒน์ ลิ้มพรจิตรวิไล, “ เรียนรู้และปฏิบัติการ ไมโครคอนโทรลเลอร์ (PIC 16F877 Microcontroller a text-lab manual) ”, กรุงเทพฯ : อินโนเวตีฟ เอ็กเพอริเมนต์, 423 หน้า



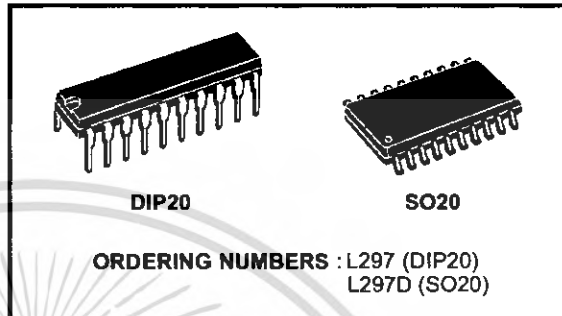


ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

STEPPER MOTOR CONTROLLERS

- NORMAL/WAVE DRIVE
- HALF/FULL STEP MODES
- CLOCKWISE/ANTICLOCKWISE DIRECTION
- SWITCHMODE LOAD CURRENT REGULATION
- PROGRAMMABLE LOAD CURRENT
- FEW EXTERNAL COMPONENTS
- RESET INPUT & HOME OUTPUT
- ENABLE INPUT



DESCRIPTION

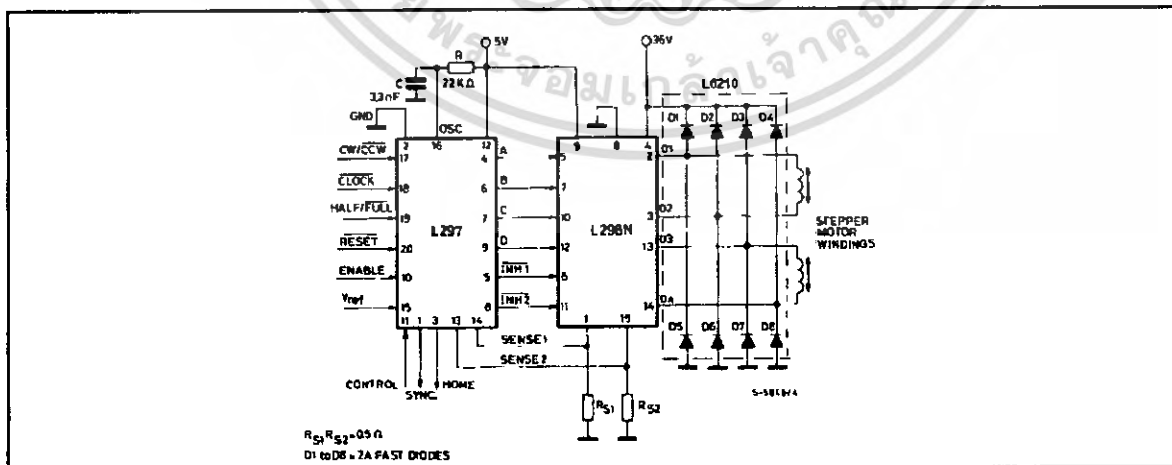
The L297/A/D Stepper Motor Controller IC generates four phase drive signals for two phase bipolar and four phase unipolar step motors in microcomputer-controlled applications. The motor can be driven in half step, normal and wave drive modes and on-chip PWM chopper circuits permit switchmode control of the current in the windings. A

feature of this device is that it requires only clock, direction and mode input signals. Since the phase are generated internally the burden on the microprocessor, and the programmer, is greatly reduced. Mounted in DIP20 and SO20 packages, the L297 can be used with monolithic bridge drives such as the L298N or L293E, or with discrete transistors and darlingtonts.

ABSOLUTE MAXIMUM RATINGS

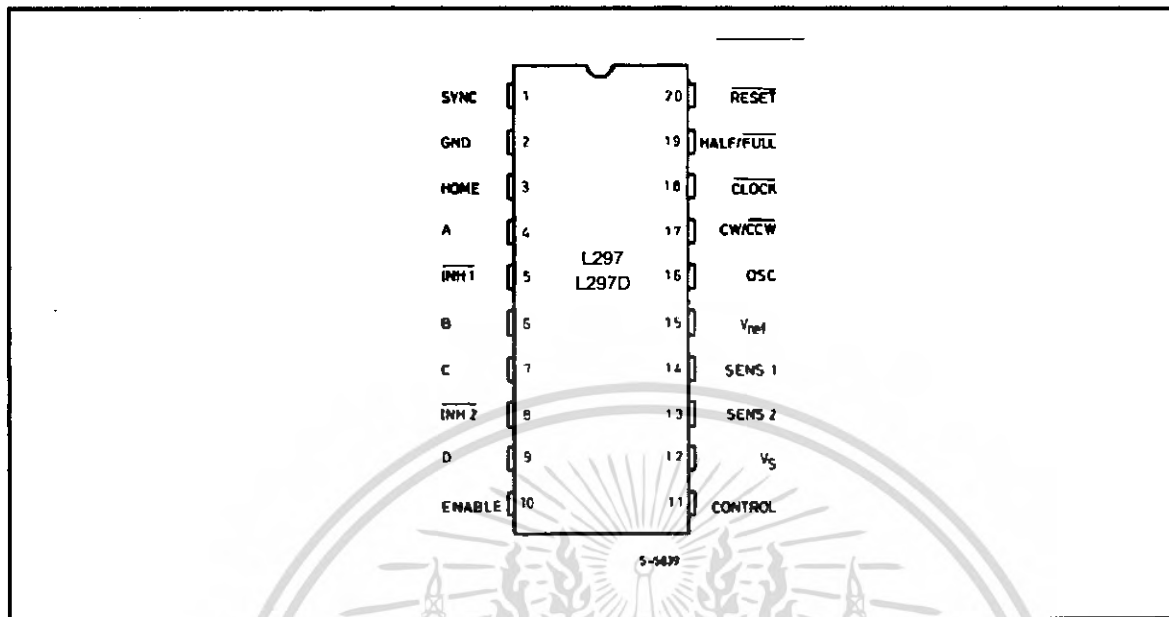
Symbol	Parameter	Value	Unit
V_s	Supply voltage	10	V
V_i	Input signals	7	V
P_{tot}	Total power dissipation ($T_{amb} = 70^\circ\text{C}$)	1	W
T_{stg}, T_j	Storage and junction temperature	-40 to + 150	$^\circ\text{C}$

TWO PHASE BIPOLAR STEPPER MOTOR CONTROL CIRCUIT

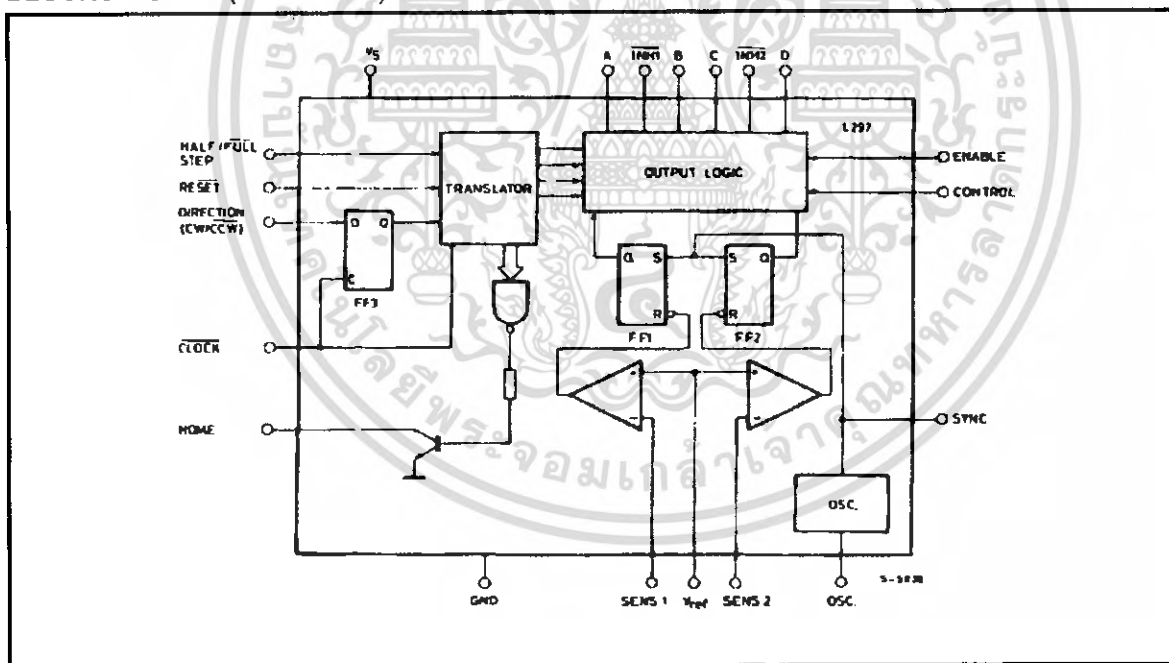


L297-L297D

PIN CONNECTION (Top view)



BLOCK DIAGRAM (L297/L297D)



PIN FUNCTIONS - L297/L297D

N°	NAME	FUNCTION
1	SYNC	Output of the on-chip chopper oscillator. The SYNC connections of all L297s to be synchronized are connected together and the oscillator components are omitted on all but one. If an external clock source is used it is injected at this terminal.
2	GND	Ground connection.
3	HOME	Open collector output that indicates when the L297 is in its initial state (ABCD = 0101). The transistor is open when this signal is active.
4	A	Motor phase A drive signal for power stage.
5	$\overline{\text{INH1}}$	Active low inhibit control for driver stage of A and B phases. When a bipolar bridge is used this signal can be used to ensure fast decay of load current when a winding is de-energized. Also used by chopper to regulate load current if CONTROL input is low.
6	B	Motor phase B drive signal for power stage.
7	C	Motor phase C drive signal for power stage.
8	$\overline{\text{INH2}}$	Active low inhibit control for drive stages of C and D phases. Same functions as INH1.
9	D	Motor phase D drive signal for power stage.
10	ENABLE	Chip enable input. When low (inactive) INH1, INH2, A, B, C and D are brought low.
11	CONTROL	Control input that defines action of chopper. When low chopper acts on INH1 and INH2; when high chopper acts on phase lines ABCD.
12	V_s	5V supply input.
13	SENS ₂	Input for load current sense voltage from power stages of phases C and D.
14	SENS ₁	Input for load current sense voltage from power stages of phases A and B.
15	V_{ref}	Reference voltage for chopper circuit. A voltage applied to this pin determines the peak load current.
16	OSC	An RC network (R to V_{CC} , C to ground) connected to this terminal determines the chopper rate. This terminal is connected to ground on all but one device in synchronized multi - L297 configurations. $f \cong 1/0.69 RC$
17	$\overline{\text{CW/CCW}}$	Clockwise/counterclockwise direction control input. Physical direction of motor rotation also depends on connection of windings. Synchronized internally therefore direction can be changed at any time.
18	$\overline{\text{CLOCK}}$	Step clock. An active low pulse on this input advances the motor one increment. The step occurs on the rising edge of this signal.

PIN FUNCTIONS - L297/L297D (continued)

N°	NAME	FUNCTION
19	HALF/FULL	Half/full step select input. When high selects half step operation, when low selects full step operation. One-phase-on full step mode is obtained by selecting FULL when the L297's translator is at an even-numbered state. Two-phase-on full step mode is set by selecting FULL when the translator is at an odd numbered position. (The home position is designate state 1).
20	RESET	Reset input. An active low pulse on this input restores the translator to the home position (state 1, ABCD = 0101).

THERMAL DATA

Symbol	Parameter	DIP20	SO20	Unit
$R_{th-j-amb}$	Thermal resistance junction-ambient	max 80	100	°C/W

CIRCUIT OPERATION

The L297 is intended for use with a dual bridge driver, quad darlington array or discrete power devices in step motor driving applications. It receives step clock, direction and mode signals from the systems controller (usually a microcomputer chip) and generates control signals for the power stage.

The principal functions are a translator, which generates the motor phase sequences, and a dual PWM chopper circuit which regulates the current in the motor windings. The translator generates three different sequences, selected by the HALF/FULL input. These are normal (two phases energised), wave drive (one phase energised), and half-step (alternately one phase energised/two phases energised). Two inhibit signals are also generated by the L297 in half step and wave drive modes. These signals, which connect directly to the L298's enable inputs, are intended to speed current decay when a winding is de-energised. When the L297 is used to drive a unipolar motor the chopper acts on these lines.

An input called CONTROL determines whether the chopper will act on the phase lines ABCD or the inhibit lines INH1 and INH2. When the phase lines

are chopped the non-active phase line of each pair (AB or CD) is activated (rather than interrupting the line then active). In L297 + L298 configurations this technique reduces dissipation in the load current sense resistors.

A common on-chip oscillator drives the dual chopper. It supplies pulses at the chopper rate which set the two flip-flops FF1 and FF2. When the current in a winding reaches the programmed peak value the voltage across the sense resistor (connected to one of the sense inputs SENS₁ or SENS₂) equals V_{ref} and the corresponding comparator resets its flip flop, interrupting the drive current until the next oscillator pulse arrives. The peak current for both windings is programmed by a voltage divider on the V_{ref} input.

Ground noise problems in multiple configurations can be avoided by synchronising the chopper oscillators. This is done by connecting all the SYNC pins together, mounting the oscillator RC network on one device only and grounding the OSC pin on all other devices.

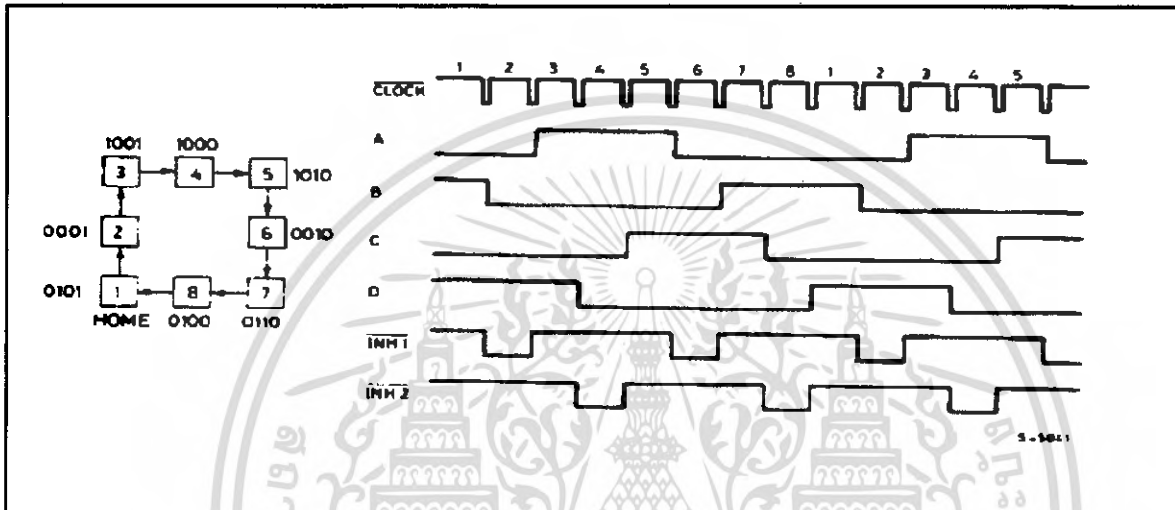
MOTOR DRIVING PHASE SEQUENCES

The L297's translator generates phase sequences for normal drive, wave drive and half step modes. The state sequences and output waveforms for these three modes are shown below. In all cases the translator advances on the low to high transition of **CLOCK**.

Clockwise rotation is indicate; for anticlockwise rotation the sequences are simply reversed **RESET** restores the translator to state 1, where ABCD = 0101.

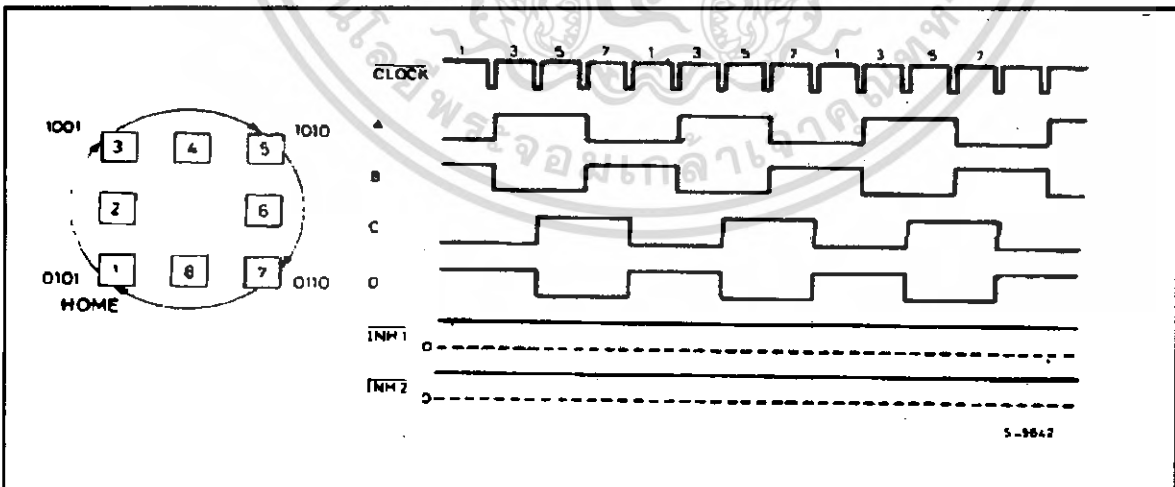
HALF STEP MODE

Half step mode is selected by a high level on the **HALF/FULL** input.



NORMAL DRIVE MODE

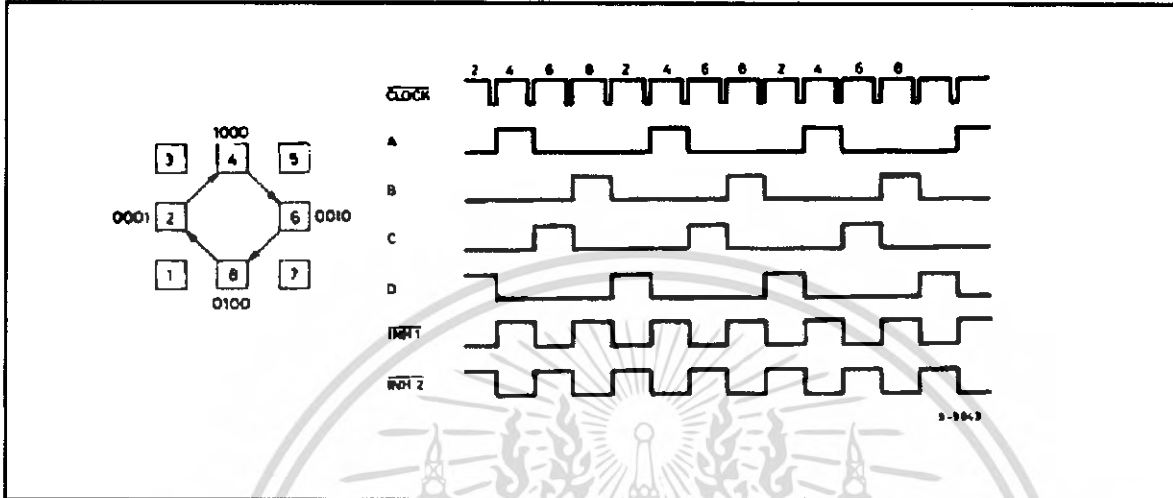
Normal drive mode (also called "two-phase-on" drive) is selected by a low level on the **HALF/FULL** input when the translator is at an odd numbered state (1, 3, 5 or 7). In this mode the **INH1** and **INH2** outputs remain high throughout.



MOTOR DRIVING PHASE SEQUENCES (continued)

WAVE DRIVE MODE

Wave drive mode (also called "one-phase-on" drive) is selected by a low level on the HALF/FULL input when the translator is at an even numbered state (2, 4, 6 or 8).



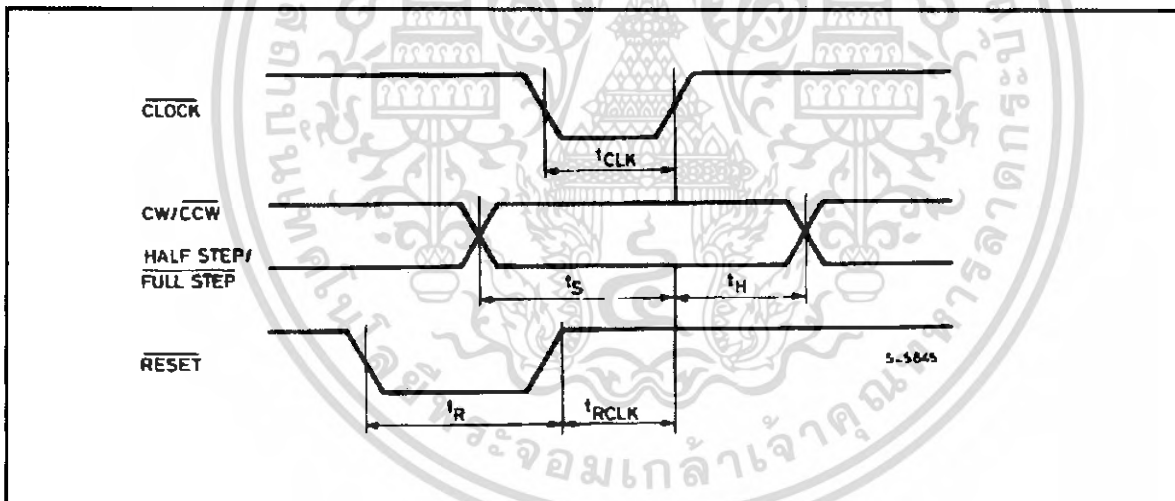
ELECTRICAL CHARACTERISTICS (Refer to the block diagram $T_{amb} = 25^{\circ}C$, $V_s = 5V$ unless otherwise specified)

Symbol	Parameter	Test conditions	Min.	Typ	Max.	Unit
V_s	Supply voltage (pin 12)		4.75		7	V
I_s	Quiescent supply current (pin 12)	Outputs floating		50	80	mA
V_i	Input voltage (pin 11, 17, 18, 19, 20)	Low			0.6	V
		High	2		V_s	V
I_i	Input current (pin 11, 17, 18, 19, 20)	$V_i = L$		100		μA
		$V_i = H$			10	μA
V_{en}	Enable input voltage (pin 10)	Low			1.3	V
		High	2		V_s	V
I_{en}	Enable input current (pin 10)	$V_{en} = L$			100	μA
		$V_{en} = H$			10	μA
V_o	Phase output voltage (pins 4, 6, 7, 9)	$I_o = 10mA$ V_{OL}			0.4	V
		$I_o = 5mA$ V_{OH}	3.9			V
V_{inh}	Inhibit output voltage (pins 5, 8)	$I_o = 10mA$ $V_{inh L}$			0.4	V
		$I_o = 5mA$ $V_{inh H}$	3.9			V
V_{sync}	Sync Output Voltage	$I_o = 5mA$ $V_{sync H}$	3.3			V
		$I_o = 5mA$ $V_{sync V}$			0.8	

ELECTRICAL CHARACTERISTICS (continued)

Symbol	Parameter	Test conditions	Min.	Typ	Max.	Unit
I_{leak}	Leakage current (pin 3)	$V_{CE} = 7V$			1	μA
V_{set}	Saturation voltage (pin 3)	$I = 5mA$			0.4	V
V_{off}	Comparators offset voltage (pins 13, 14, 15)	$V_{ref} = 1V$			5	mV
I_o	Comparator bias current (pins 13, 14, 15)		-100		10	μA
V_{ref}	Input reference voltage (pin 15)		0		3	V
t_{CLK}	Clock time		0.5			μs
t_s	Set up time		1			μs
t_H	Hold time		4			μs
t_R	Reset time		1			μs
t_{RCLK}	Reset to clock delay		1			μs

Figure 1.



APPLICATION INFORMATION

TWO PHASE BIPOLAR STEPPER MOTOR CONTROL CIRCUIT

This circuit drives bipolar stepper motors with winding currents up to 2A. The diodes are fast 2A types.

Figure 2.

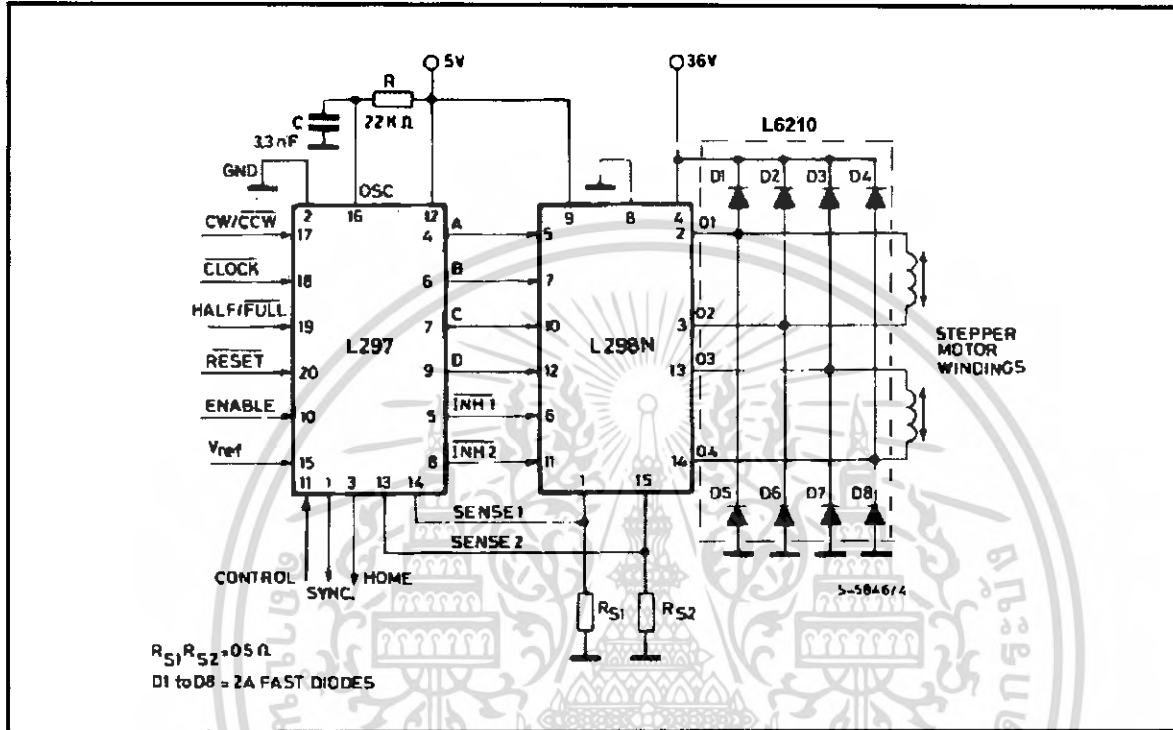
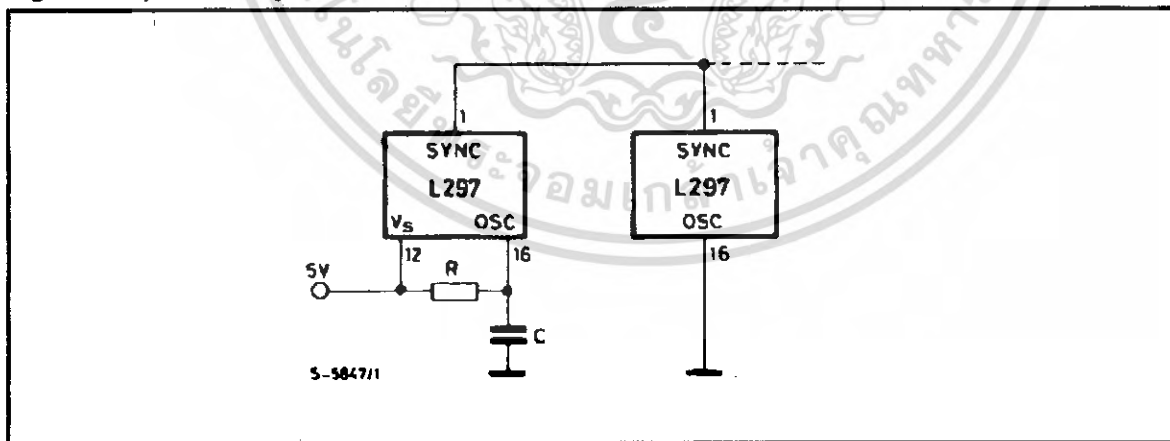


Figure 3 : Synchronising L297s

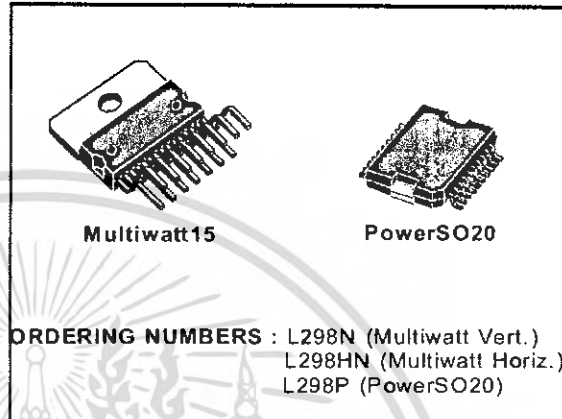


DUAL FULL-BRIDGE DRIVER

- OPERATING SUPPLY VOLTAGE UP TO 46 V
- TOTAL DC CURRENT UP TO 4 A
- LOW SATURATION VOLTAGE
- OVERTEMPERATURE PROTECTION
- LOGICAL "0" INPUT VOLTAGE UP TO 1.5 V (HIGH NOISE IMMUNITY)

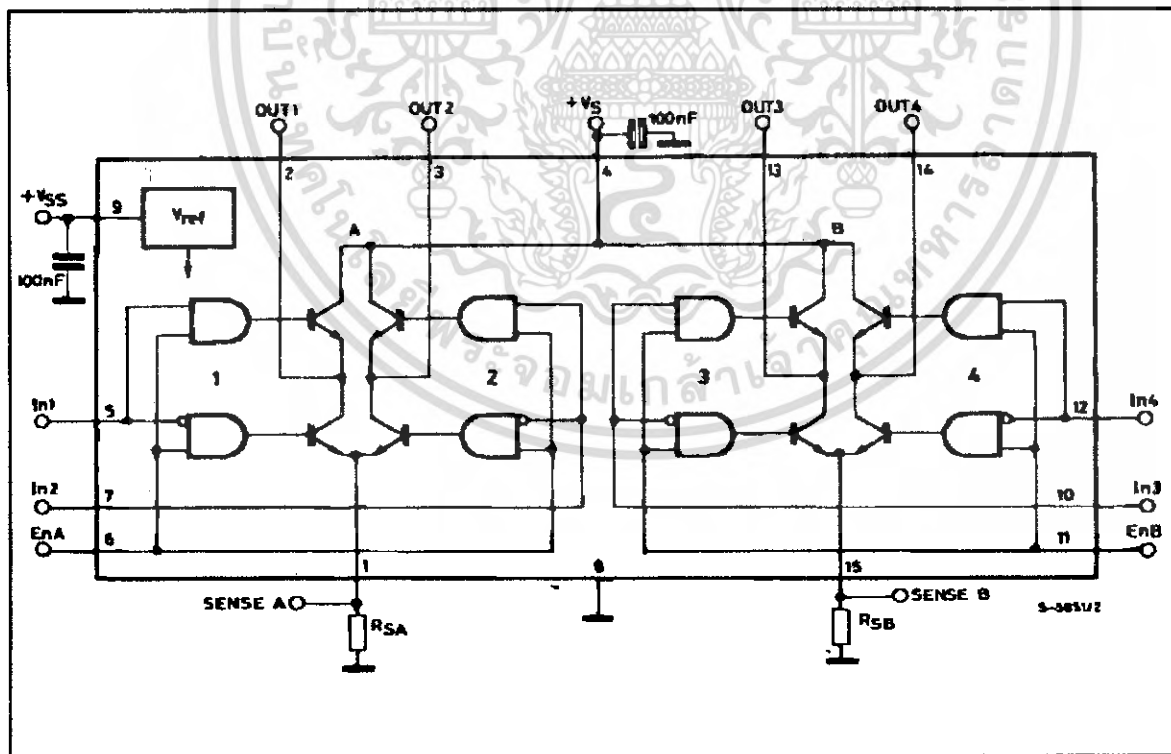
DESCRIPTION

The L298 is an integrated monolithic circuit in a 15-lead Multiwatt and PowerSO20 packages. It is a high voltage, high current dual full-bridge driver designed to accept standard TTL logic levels and drive inductive loads such as relays, solenoids, DC and stepping motors. Two enable inputs are provided to enable or disable the device independently of the input signals. The emitters of the lower transistors of each bridge are connected together and the corresponding external terminal can be used for the con-



nection of an external sensing resistor. An additional supply input is provided so that the logic works at a lower voltage.

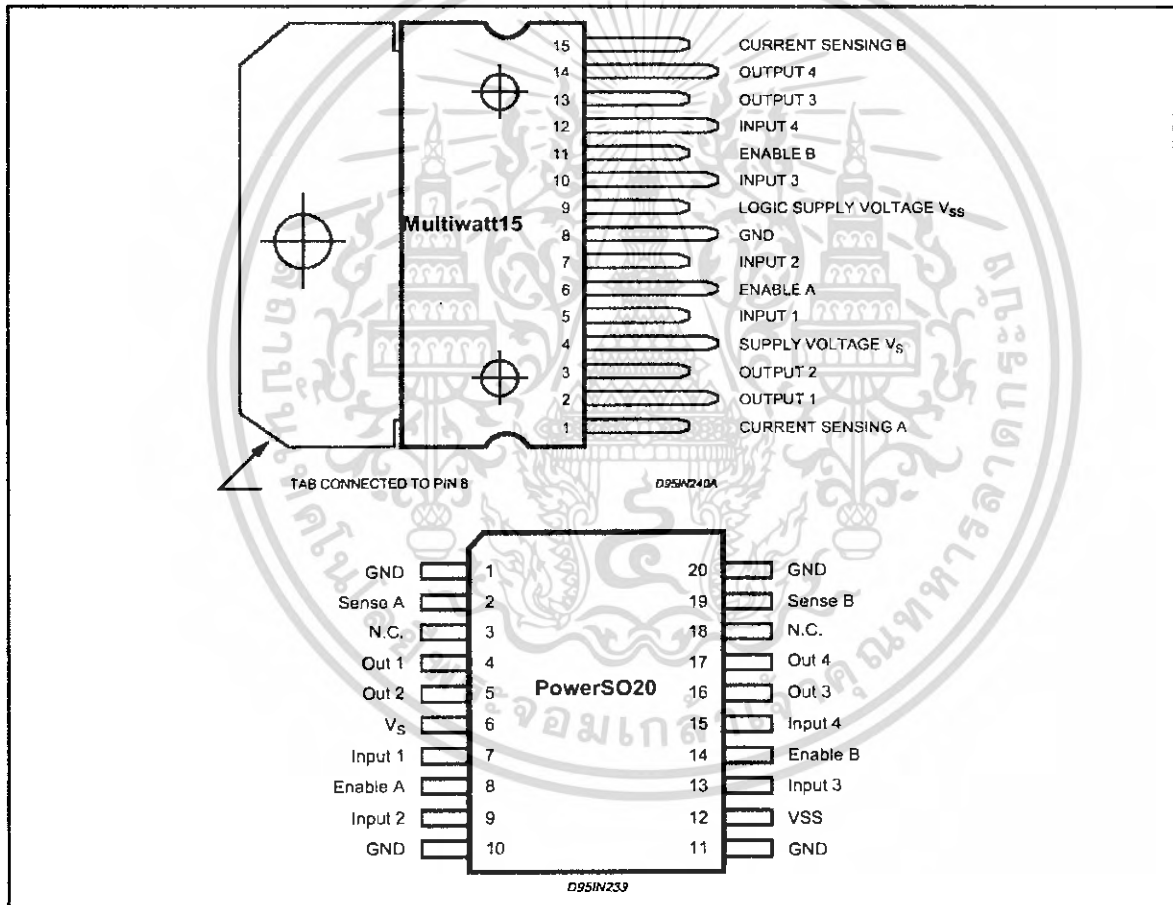
BLOCK DIAGRAM



ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Value	Unit
V_S	Power Supply	50	V
V_{SS}	Logic Supply Voltage	7	V
V_I, V_{En}	Input and Enable Voltage	-0.3 to 7	V
I_O	Peak Output Current (each Channel)		
	- Non Repetitive ($t = 100\mu s$)	3	A
	- Repetitive (80% on -20% off; $t_{on} = 10ms$)	2.5	A
	-DC Operation	2	A
V_{sens}	Sensing Voltage	-1 to 2.3	V
P_{tot}	Total Power Dissipation ($T_{case} = 75^\circ C$)	25	W
T_{op}	Junction Operating Temperature	-25 to 130	$^\circ C$
T_{stg}, T_j	Storage and Junction Temperature	-40 to 150	$^\circ C$

PIN CONNECTIONS (top view)



THERMAL DATA

Symbol	Parameter		PowerSO20	Multiwatt15	Unit
$R_{th(j-case)}$	Thermal Resistance Junction-case	Max.	-	3	$^\circ C/W$
$R_{th(j-amb)}$	Thermal Resistance Junction-ambient	Max.	13 (*)	35	$^\circ C/W$

(*) Mounted on aluminum substrate

PIN FUNCTIONS (refer to the block diagram)

MW.15	PowerSO	Name	Function
1;15	2;19	Sense A; Sense B	Between this pin and ground is connected the sense resistor to control the current of the load.
2;3	4;5	Out 1; Out 2	Outputs of the Bridge A; the current that flows through the load connected between these two pins is monitored at pin 1.
4	6	V _S	Supply Voltage for the Power Output Stages. A non-inductive 100nF capacitor must be connected between this pin and ground.
5;7	7;9	Input 1; Input 2	TTL Compatible Inputs of the Bridge A.
6;11	8;14	Enable A; Enable B	TTL Compatible Enable Input; the L state disables the bridge A (enable A) and/or the bridge B (enable B).
8	1,10,11,20	GND	Ground.
9	12	V _{SS}	Supply Voltage for the Logic Blocks. A100nF capacitor must be connected between this pin and ground.
10; 12	13;15	Input 3; Input 4	TTL Compatible Inputs of the Bridge B.
13; 14	16;17	Out 3; Out 4	Outputs of the Bridge B. The current that flows through the load connected between these two pins is monitored at pin 15.
-	3;18	N.C.	Not Connected

ELECTRICAL CHARACTERISTICS (V_S = 42V; V_{SS} = 5V; T_J = 25°C; unless otherwise specified)

Symbol	Parameter	Test Conditions	Min.	Typ.	Max.	Unit
V _S	Supply Voltage (pin 4)	Operative Condition	V _{IH} +2.5		46	V
V _{SS}	Logic Supply Voltage (pin 9)		4.5	5	7	V
I _S	Quiescent Supply Current (pin 4)	V _{en} = H; I _L = 0	V _i = L V _i = H	13 50	22 70	mA mA
		V _{en} = L	V _i = X		4	mA
I _{SS}	Quiescent Current from V _{SS} (pin 9)	V _{en} = H; I _L = 0	V _i = L V _i = H	24 7	36 12	mA mA
		V _{en} = L	V _i = X		6	mA
V _{IL}	Input Low Voltage (pins 5, 7, 10, 12)		-0.3		1.5	V
V _{IH}	Input High Voltage (pins 5, 7, 10, 12)		2.3		V _{SS}	V
I _{IL}	Low Voltage Input Current (pins 5, 7, 10, 12)	V _i = L			-10	μA
I _{IH}	High Voltage Input Current (pins 5, 7, 10, 12)	V _i = H ≤ V _{SS} - 0.6V		30	100	μA
V _{en} = L	Enable Low Voltage (pins 6, 11)		-0.3		1.5	V
V _{en} = H	Enable High Voltage (pins 6, 11)		2.3		V _{SS}	V
I _{en} = L	Low Voltage Enable Current (pins 6, 11)	V _{en} = L			-10	μA
I _{en} = H	High Voltage Enable Current (pins 6, 11)	V _{en} = H ≤ V _{SS} - 0.6V		30	100	μA
V _{CEsat(H)}	Source Saturation Voltage	I _L = 1A I _L = 2A	0.95	1.35 2	1.7 2.7	V V
V _{CEsat(L)}	Sink Saturation Voltage	I _L = 1A (5) I _L = 2A (5)	0.85	1.2 1.7	1.6 2.3	V V
V _{CEsat}	Total Drop	I _L = 1A (5) I _L = 2A (5)	1.80		3.2 4.9	V V
V _{sens}	Sensing Voltage (pins 1, 15)		-1 (1)		2	V

ELECTRICAL CHARACTERISTICS (continued)

Symbol	Parameter	Test Conditions	Min.	Typ.	Max.	Unit
T ₁ (V _i)	Source Current Turn-off Delay	0.5 V _i to 0.9 I _L (2); (4)		1.5		μs
T ₂ (V _i)	Source Current Fall Time	0.9 I _L to 0.1 I _L (2); (4)		0.2		μs
T ₃ (V _i)	Source Current Turn-on Delay	0.5 V _i to 0.1 I _L (2); (4)		2		μs
T ₄ (V _i)	Source Current Rise Time	0.1 I _L to 0.9 I _L (2); (4)		0.7		μs
T ₅ (V _i)	Sink Current Turn-off Delay	0.5 V _i to 0.9 I _L (3); (4)		0.7		μs
T ₆ (V _i)	Sink Current Fall Time	0.9 I _L to 0.1 I _L (3); (4)		0.25		μs
T ₇ (V _i)	Sink Current Turn-on Delay	0.5 V _i to 0.9 I _L (3); (4)		1.6		μs
T ₈ (V _i)	Sink Current Rise Time	0.1 I _L to 0.9 I _L (3); (4)		0.2		μs
f _c (V _i)	Commutation Frequency	I _L = 2A		25	40	KHz
T ₁ (V _{en})	Source Current Turn-off Delay	0.5 V _{en} to 0.9 I _L (2); (4)		3		μs
T ₂ (V _{en})	Source Current Fall Time	0.9 I _L to 0.1 I _L (2); (4)		1		μs
T ₃ (V _{en})	Source Current Turn-on Delay	0.5 V _{en} to 0.1 I _L (2); (4)		0.3		μs
T ₄ (V _{en})	Source Current Rise Time	0.1 I _L to 0.9 I _L (2); (4)		0.4		μs
T ₅ (V _{en})	Sink Current Turn-off Delay	0.5 V _{en} to 0.9 I _L (3); (4)		2.2		μs
T ₆ (V _{en})	Sink Current Fall Time	0.9 I _L to 0.1 I _L (3); (4)		0.35		μs
T ₇ (V _{en})	Sink Current Turn-on Delay	0.5 V _{en} to 0.9 I _L (3); (4)		0.25		μs
T ₈ (V _{en})	Sink Current Rise Time	0.1 I _L to 0.9 I _L (3); (4)		0.1		μs

- 1) Sensing voltage can be -1 V for t ≤ 50 μsec; in steady state V_{sens} min ≥ -0.5 V.
- 2) See fig. 2.
- 3) See fig. 4.
- 4) The load must be a pure resistor.

Figure 1 : Typical Saturation Voltage vs. Output Current.

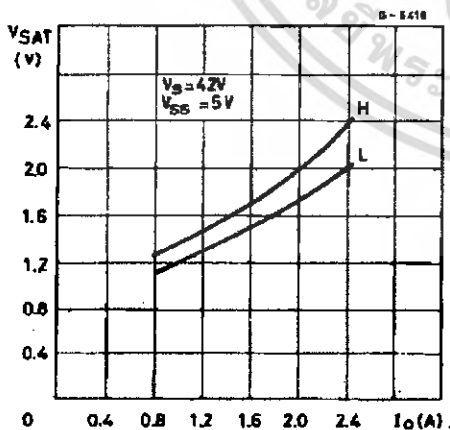
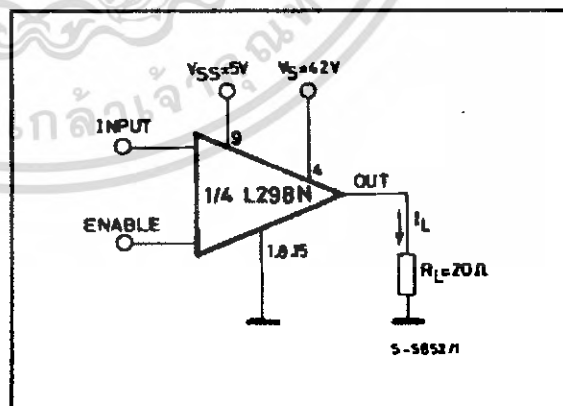


Figure 2 : Switching Times Test Circuits.



Note : For INPUT Switching, set EN = H
For ENABLE Switching, set IN = H



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Figure 3 : Source Current Delay Times vs. Input or Enable Switching.

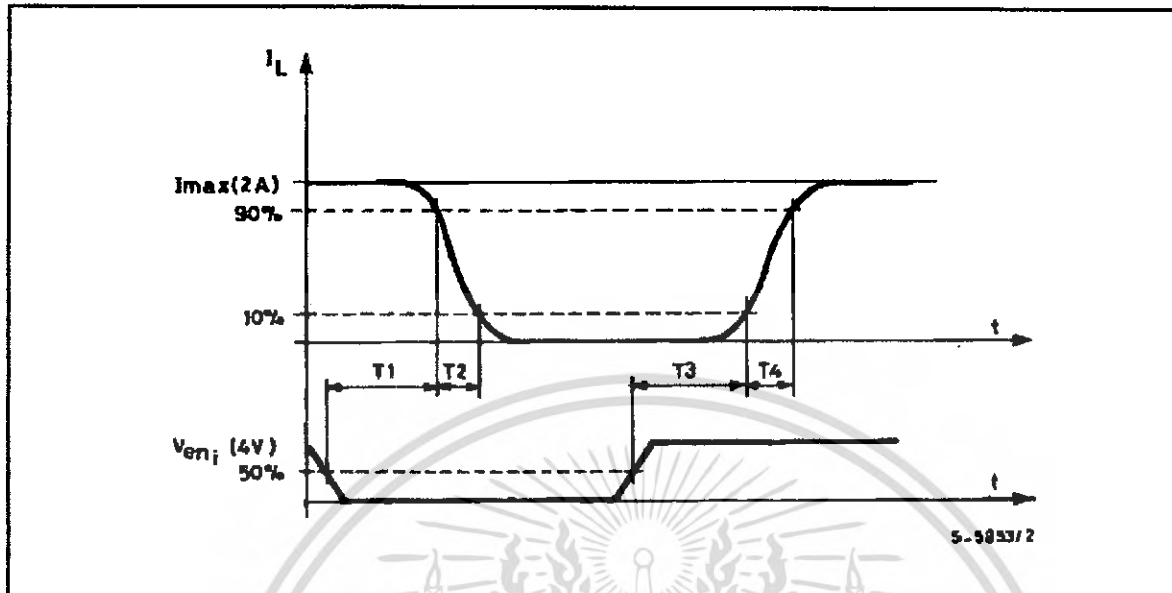
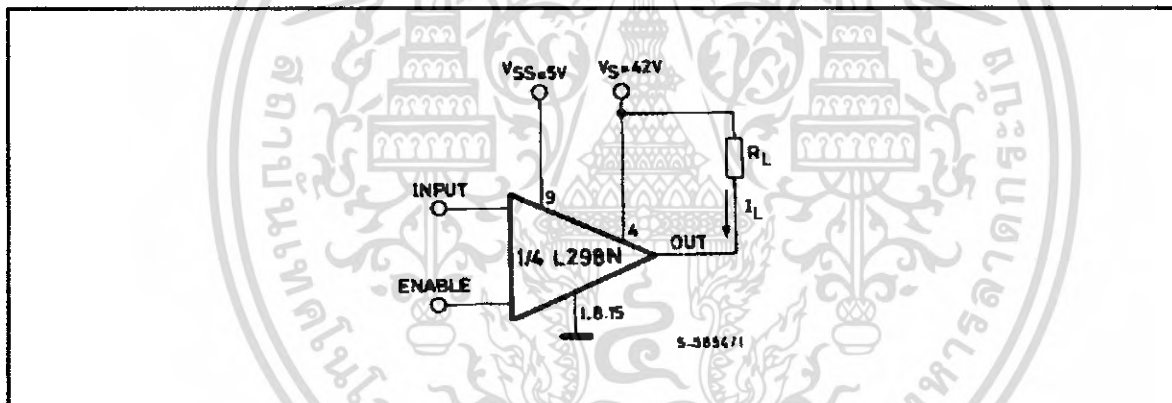


Figure 4 : Switching Times Test Circuits.



Note : For INPUT Switching, set EN = H
 For ENABLE Switching, set IN = L

Figure 5 : Sink Current Delay Times vs. Input 0 V Enable Switching.

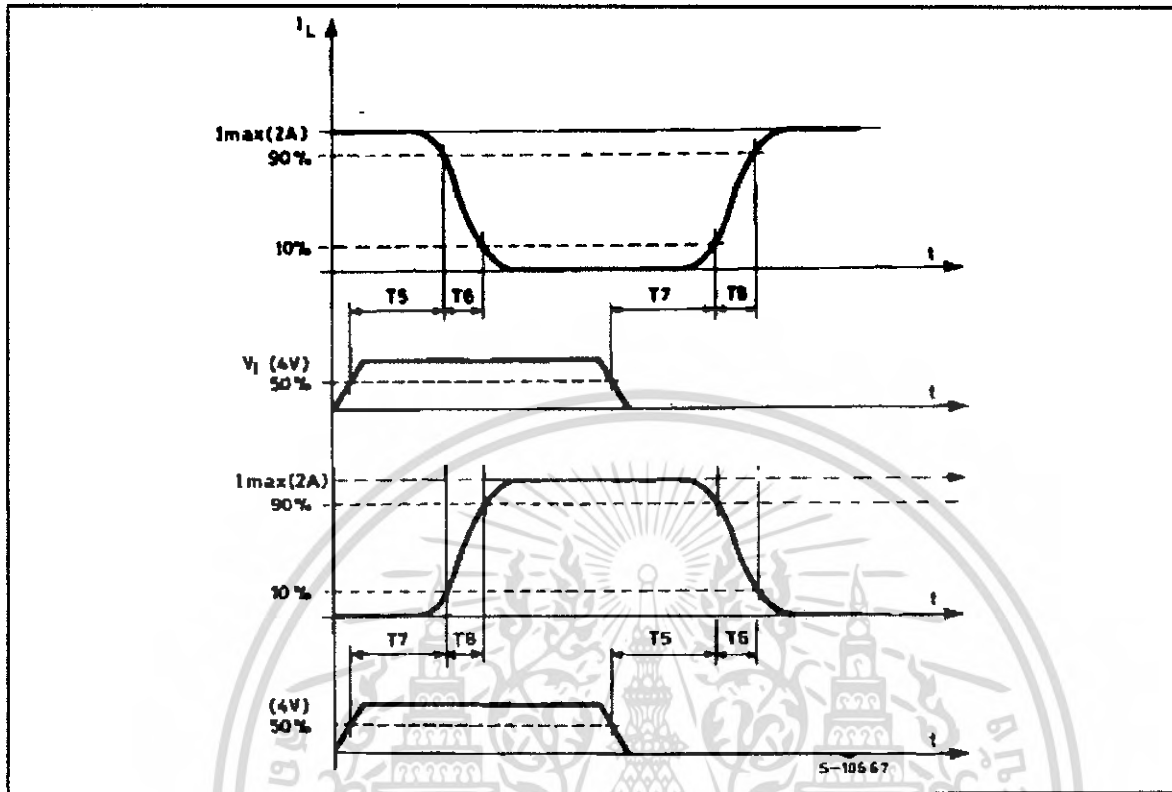
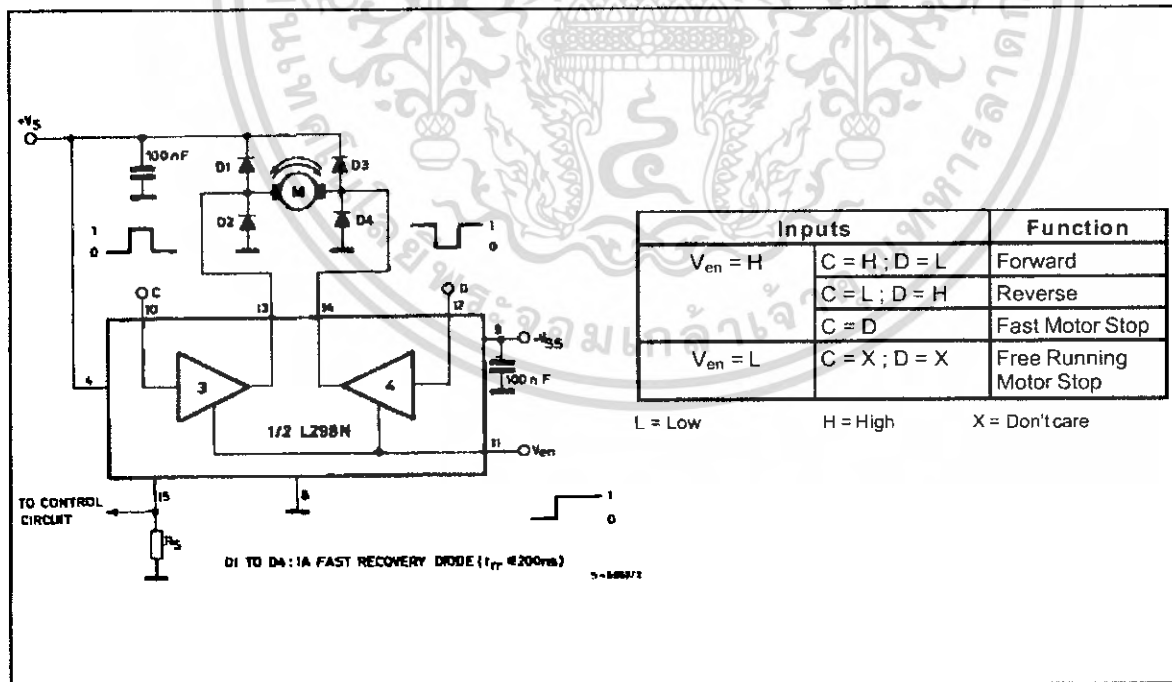
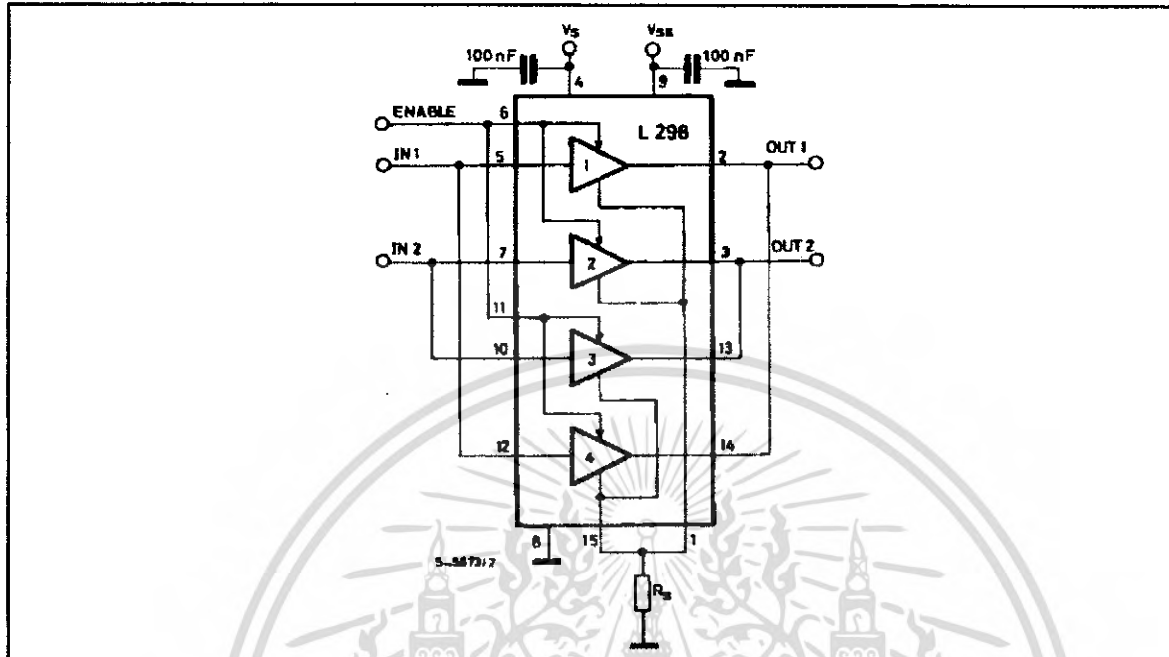


Figure 6 : Bidirectional DC Motor Control.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Figure 7 : For higher currents, outputs can be paralleled. Take care to parallel channel 1 with channel 4 and channel 2 with channel 3.



APPLICATION INFORMATION (Refer to the block diagram)

1.1. POWER OUTPUT STAGE

The L298 integrates two power output stages (A; B). The power output stage is a bridge configuration and its outputs can drive an inductive load in common or differential mode, depending on the state of the inputs. The current that flows through the load comes out from the bridge at the sense output: an external resistor (R_{SA} ; R_{SB}) allows to detect the intensity of this current.

1.2. INPUT STAGE

Each bridge is driven by means of four gates the input of which are $In1$; $In2$; EnA and $In3$; $In4$; EnB . The In inputs set the bridge state when The En input is high; a low state of the En input inhibits the bridge. All the inputs are TTL compatible.

2. SUGGESTIONS

A non inductive capacitor, usually of 100 nF, must be foreseen between both V_s and V_{ss} , to ground, as near as possible to GND pin. When the large capacitor of the power supply is too far from the IC, a second smaller one must be foreseen near the L298.

The sense resistor, not of a wire wound type, must be grounded near the negative pole of V_s that must be near the GND pin of the I.C.

Each input must be connected to the source of the driving signals by means of a very short path.

Turn-On and Turn-Off : Before to Turn-ON the Supply Voltage and before to Turn it OFF, the Enable input must be driven to the Low state.

3. APPLICATIONS

Fig 6 shows a bidirectional DC motor control Schematic Diagram for which only one bridge is needed. The external bridge of diodes $D1$ to $D4$ is made by four fast recovery elements ($tr \leq 200$ nsec) that must be chosen of a V_F as low as possible at the worst case of the load current.

The sense output voltage can be used to control the current amplitude by chopping the inputs, or to provide overcurrent protection by switching low the enable input.

The brake function (Fast motor stop) requires that the Absolute Maximum Rating of 2 Amps must never be overcome.

When the repetitive peak current needed from the load is higher than 2 Amps, a paralleled configuration can be chosen (See Fig. 7).

An external bridge of diodes are required when inductive loads are driven and when the inputs of the IC are chopped; Schottky diodes would be preferred.

L298

This solution can drive until 3 Amps In DC operation and until 3.5 Amps of a repetitive peak current.

On Fig 8 it is shown the driving of a two phase bipolar stepper motor ; the needed signals to drive the inputs of the L298 are generated, in this example, from the IC L297.

Fig 9 shows an example of P.C.B. designed for the application of Fig 8.

Figure 8 : Two Phase Bipolar Stepper Motor Circuit.

This circuit drives bipolar stepper motors with winding currents up to 2 A. The diodes are fast 2 A types.

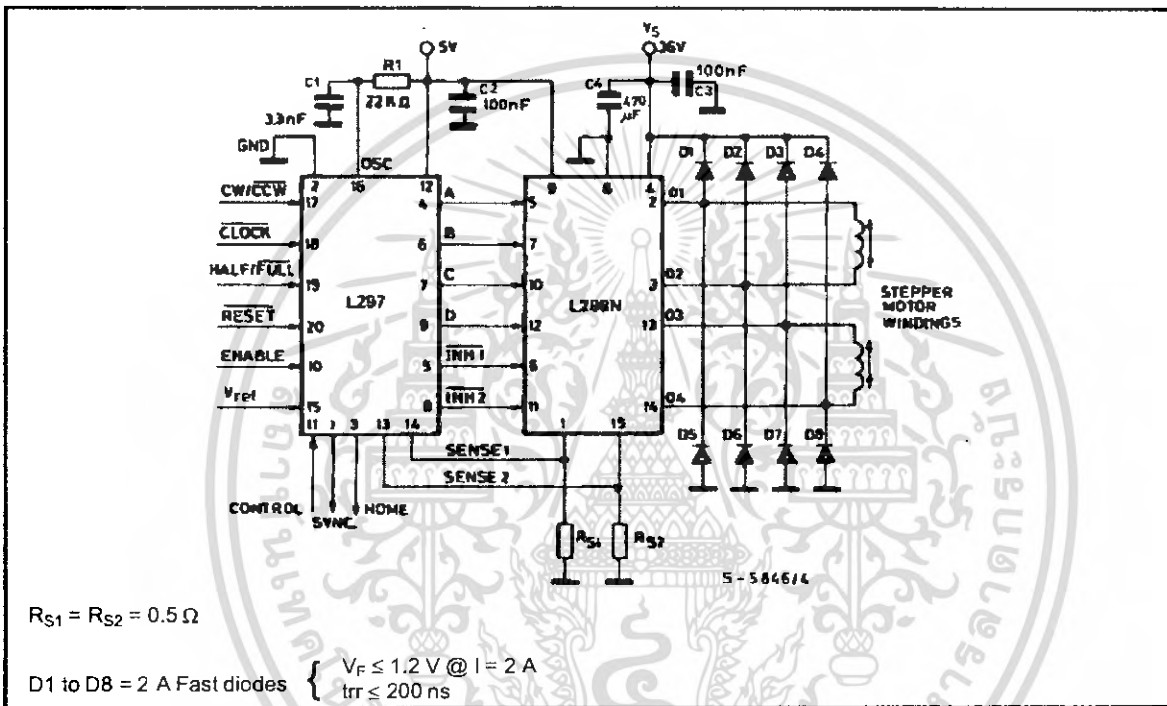


Fig 10 shows a second two phase bipolar stepper motor control circuit where the current is controlled by the I.C. L6506.

PC817 Series

High Density Mounting Type Photocoupler

- Lead forming type (I type) and taping reel type (P type) are also available. (PC817/PC817P)
- TÜV (VDE0884) approved type is also available as an option.

■ Features

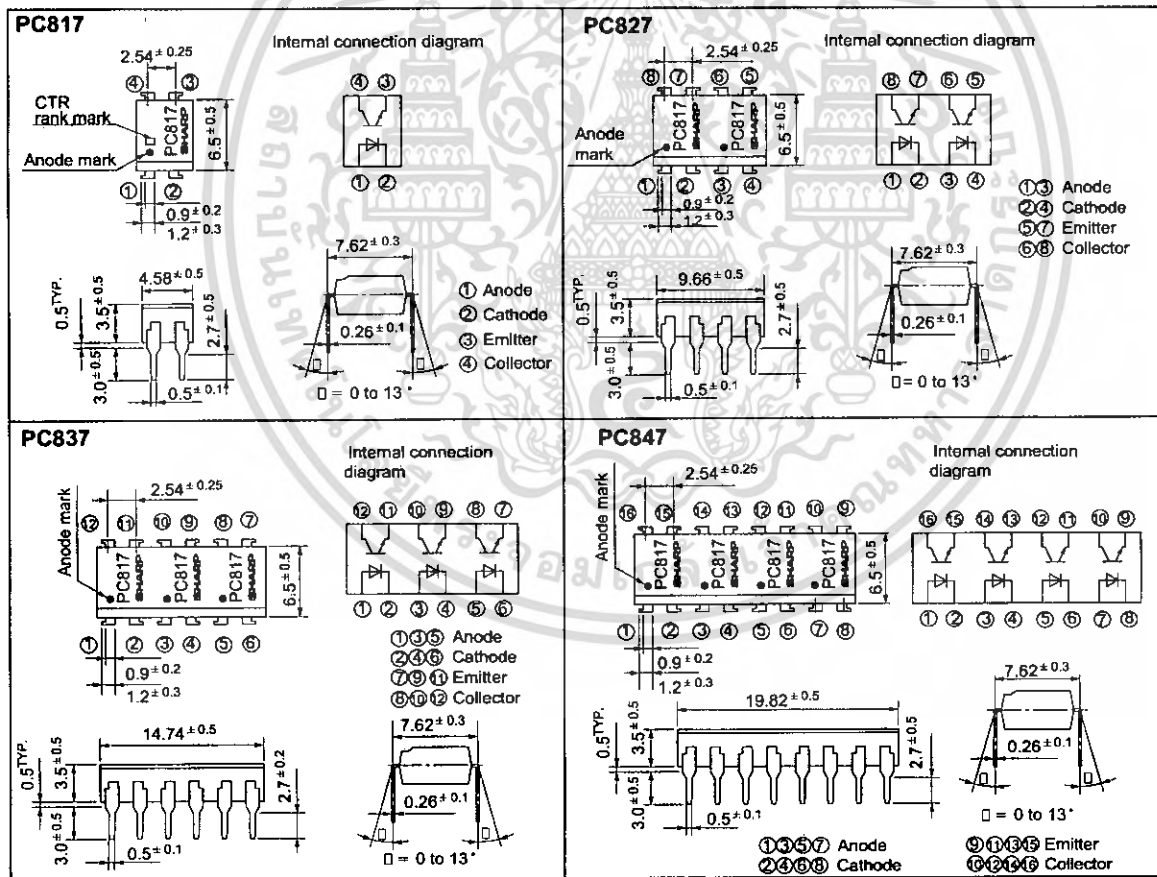
1. Current transfer ratio
(CTR: MIN. 50% at $I_F = 5\text{mA}$, $V_{CE} = 5\text{V}$)
2. High isolation voltage between input and output ($V_{iso} : 5000\text{V}_{rms}$)
3. Compact dual-in-line package
PC817 : 1-channel type
PC827 : 2-channel type
PC837 : 3-channel type
PC847 : 4-channel type
4. Recognized by UL, file No. E64380

■ Applications

1. Computer terminals
2. System appliances, measuring instruments
3. Registers, copiers, automatic vending machines
4. Electric home appliances, such as fan heaters, etc.
5. Signal transmission between circuits of different potentials and impedances

■ Outline Dimensions

(Unit : mm)



* In the absence of confirmation by device specification sheets, SHARP takes no responsibility for any defects that occur in equipment using any of SHARP's devices, shown in catalogs, data books, etc. Contact SHARP in order to obtain the latest version of the device specification sheets before using any SHARP's device."

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

■ Absolute Maximum Ratings (Ta= 25°C)

Parameter	Symbol	Rating	Unit
Input	Forward current	I_F	50 mA
	*1 Peak forward current	I_{FM}	1 A
	Reverse voltage	V_R	6 V
	Power dissipation	P	70 mW
Output	Collector-emitter voltage	V_{CEO}	35 V
	Emitter-collector voltage	V_{ECO}	6 V
	Collector current	I_C	50 mA
	Collector power dissipation	P_C	150 mW
	Total power dissipation	P_{tot}	200 mW
	*2 Isolation voltage	V_{iso}	5 000 V_{rms}
Operating temperature	T_{opr}	- 30 to + 100	°C
Storage temperature	T_{stg}	- 55 to + 125	°C
*3 Soldering temperature	T_{sol}	260	°C

*1 Pulse width $\leq 100\mu s$, Duty ratio : 0.001

*2 40 to 60% RH, AC for 1 minute

*3 For 10 seconds

■ Electro-optical Characteristics (Ta= 25°C)

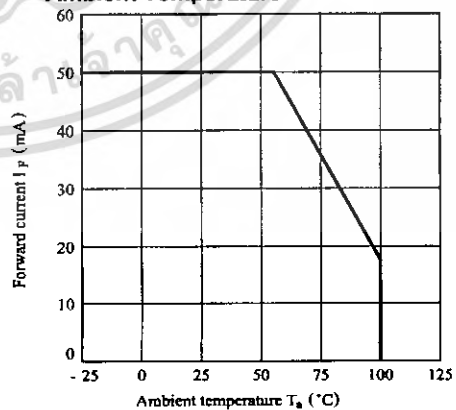
Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit	
Input	Forward voltage	V_F $I_F = 20mA$	-	1.2	1.4	V	
	Peak forward voltage	V_{FM} $I_{FM} = 0.5A$	-	-	3.0	V	
	Reverse current	I_R $V_R = 4V$	-	-	10	μA	
	Terminal capacitance	C_t $V = 0, f = 1kHz$	-	30	250	pF	
Output	Collector dark current	I_{CBO} $V_{CE} = 20V$	-	-	10^{-7}	A	
Transfer characteristics	*4 Current transfer ratio	CTR $I_F = 5mA, V_{CE} = 5V$	50	-	600	%	
	Collector-emitter saturation voltage	$V_{CE(sat)}$ $I_F = 20mA, I_C = 1mA$	-	0.1	0.2	V	
	Isolation resistance	R_{iso} DC500V, 40 to 60% RH	5×10^{10}	10^{11}	-	Ω	
	Floating capacitance	C_f $V = 0, f = 1MHz$	-	0.6	1.0	pF	
	Response time	Cut-off frequency	f_c $V_{CE} = 5V, I_C = 2mA, R_L = 100\Omega, -3dB$	-	80	-	kHz
			Rise time t_r $V_{CE} = 2V, I_C = 2mA, R_L = 100\Omega$	-	4	18	μs
	Fall time t_f		-	3	18	μs	

*4 Classification table of current transfer ratio is shown below.

Model No.	Rank mark	CTR (%)
PC817A	A	80 to 160
PC817B	B	130 to 260
PC817C	C	200 to 400
PC817D	D	300 to 600
PC8●7AB	A or B	80 to 260
PC8●7BC	B or C	130 to 400
PC8●7CD	C or D	200 to 600
PC8●7AC	A, B or C	80 to 400
PC8●7BD	B, C or D	130 to 600
PC8●7AD	A, B, C or D	80 to 600
PC8●7	A, B, C, D or No mark	50 to 600

● : 1 or 2 or 3 or 4

Fig. 1 Forward Current vs. Ambient Temperature



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Fig. 2 Collector Power Dissipation vs. Ambient Temperature

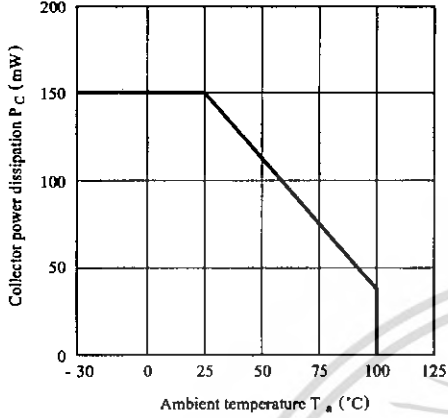


Fig. 3 Peak Forward Current vs. Duty Ratio

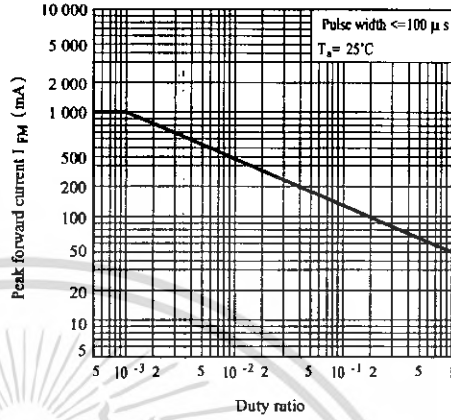


Fig. 4 Current Transfer Ratio vs. Forward Current

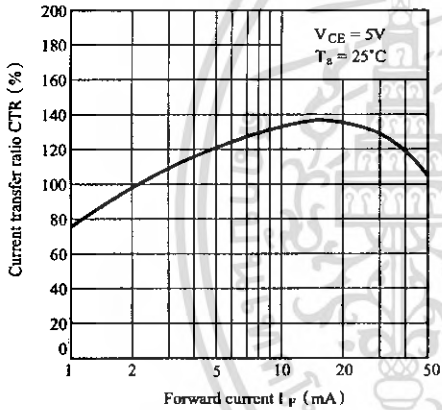


Fig. 5 Forward Current vs. Forward Voltage

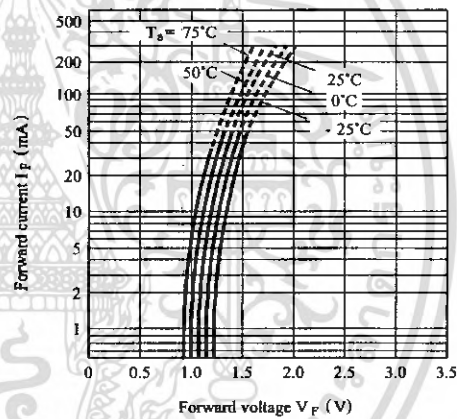


Fig. 6 Collector Current vs. Collector-emitter Voltage

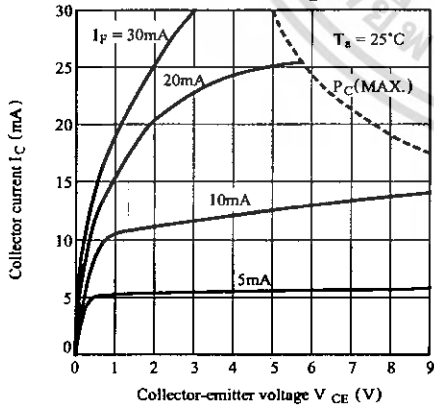
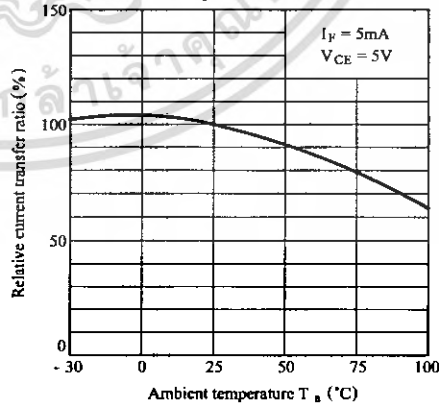


Fig. 7 Relative Current Transfer Ratio vs. Ambient Temperature



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Fig. 8 Collector-emitter Saturation Voltage vs. Ambient Temperature

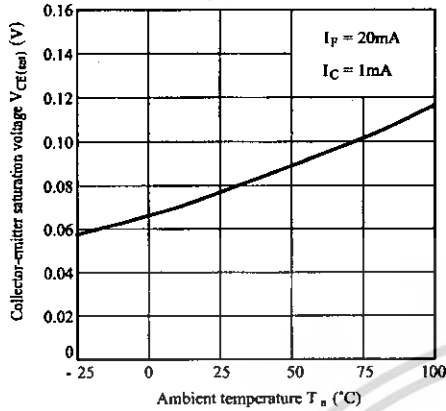


Fig. 9 Collector Dark Current vs. Ambient Temperature

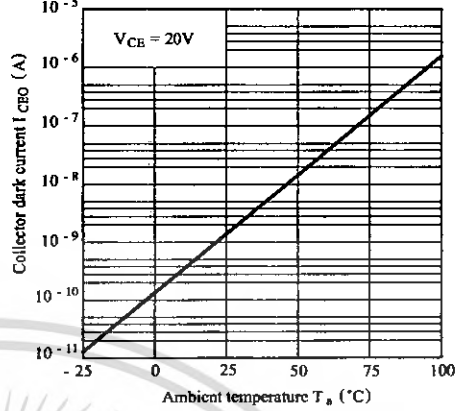


Fig.10 Response Time vs. Load Resistance

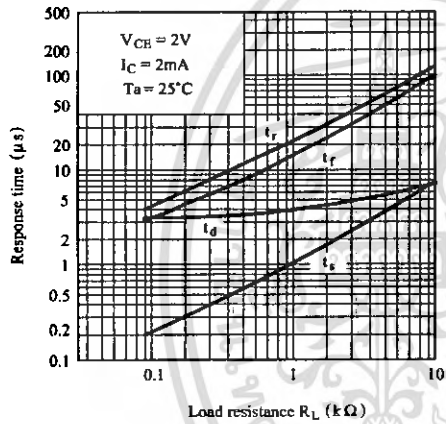
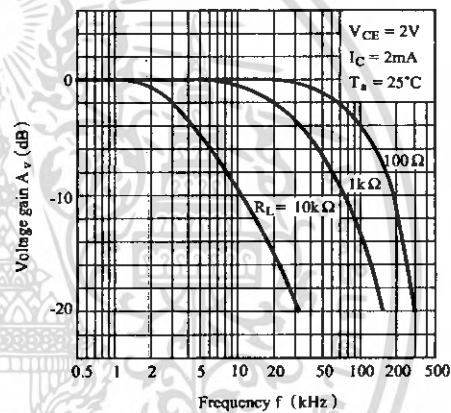
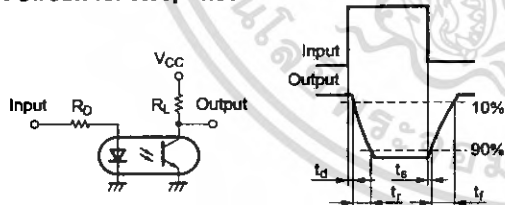


Fig.11 Frequency Response



Test Circuit for Response Time



Test Circuit for Frequency Response

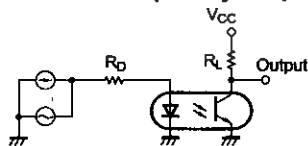
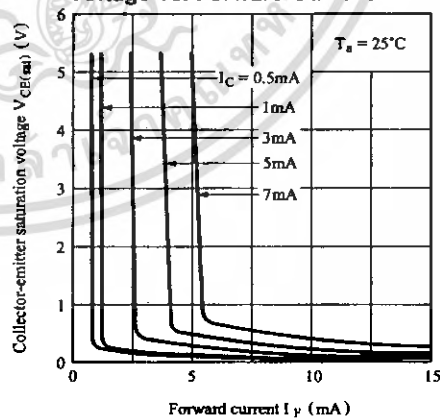


Fig.12 Collector-emitter Saturation Voltage vs. Forward Current



● Please refer to the chapter "Precautions for Use"

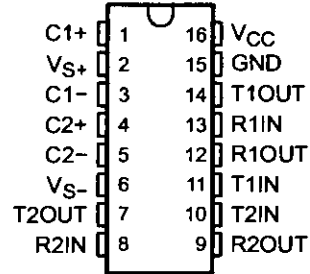
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MAX232, MAX232I DUAL EIA-232 DRIVERS/RECEIVERS

SLLS047L – FEBRUARY 1989 – REVISED MARCH 2004

- Meets or Exceeds TIA/EIA-232-F and ITU Recommendation V.28
- Operates From a Single 5-V Power Supply With 1.0- μ F Charge-Pump Capacitors
- Operates Up To 120 kbit/s
- Two Drivers and Two Receivers
- ± 30 -V Input Levels
- Low Supply Current . . . 8 mA Typical
- ESD Protection Exceeds JESD 22 – 2000-V Human-Body Model (A114-A)
- Upgrade With Improved ESD (15-kV HBM) and 0.1- μ F Charge-Pump Capacitors is Available With the MAX202
- Applications
 - TIA/EIA-232-F, Battery-Powered Systems, Terminals, Modems, and Computers

MAX232 . . . D, DW, N, OR NS PACKAGE
MAX232I . . . D, DW, OR N PACKAGE
(TOP VIEW)



description/ordering information

The MAX232 is a dual driver/receiver that includes a capacitive voltage generator to supply TIA/EIA-232-F voltage levels from a single 5-V supply. Each receiver converts TIA/EIA-232-F inputs to 5-V TTL/CMOS levels. These receivers have a typical threshold of 1.3 V, a typical hysteresis of 0.5 V, and can accept ± 30 -V inputs. Each driver converts TTL/CMOS input levels into TIA/EIA-232-F levels. The driver, receiver, and voltage-generator functions are available as cells in the Texas Instruments LinASIC™ library.

ORDERING INFORMATION

TA	PACKAGE†		ORDERABLE PART NUMBER	TOP-SIDE MARKING
0°C to 70°C	PDIP (N)	Tube of 25	MAX232N	MAX232N
	SOIC (D)	Tube of 40	MAX232D	MAX232
		Reel of 2500	MAX232DR	
	SOIC (DW)	Tube of 40	MAX232DW	MAX232
		Reel of 2000	MAX232DWR	
SOP (NS)	Reel of 2000	MAX232NSR	MAX232	
-40°C to 85°C	PDIP (N)	Tube of 25	MAX232IN	MAX232IN
	SOIC (D)	Tube of 40	MAX232ID	MAX232I
		Reel of 2500	MAX232IDR	
	SOIC (DW)	Tube of 40	MAX232IDW	MAX232I
		Reel of 2000	MAX232IDWR	

† Package drawings, standard packing quantities, thermal data, symbolization, and PCB design guidelines are available at www.ti.com/sc/package.



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

LinASIC is a trademark of Texas Instruments.

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

Copyright © 2004, Texas Instruments Incorporated

**TEXAS
INSTRUMENTS**

POST OFFICE BOX 655303 • DALLAS, TEXAS 75265

1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MAX232, MAX232I DUAL EIA-232 DRIVERS/RECEIVERS

SLLS047L - FEBRUARY 1989 - REVISED MARCH 2004

Function Tables

EACH DRIVER

INPUT TIN	OUTPUT TOUT
L	H
H	L

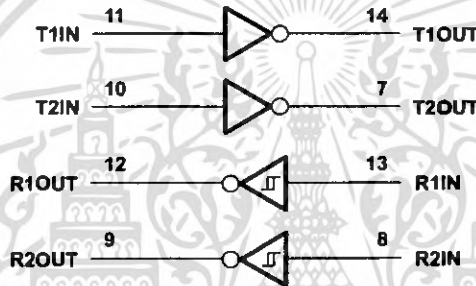
H = high level, L = low level

EACH RECEIVER

INPUT RIN	OUTPUT ROUT
L	H
H	L

H = high level, L = low level

logic diagram (positive logic)



MAX232, MAX232I DUAL EIA-232 DRIVERS/RECEIVERS

SLLS047L - FEBRUARY 1989 - REVISED MARCH 2004

absolute maximum ratings over operating free-air temperature range (unless otherwise noted)†

Input supply voltage range, V_{CC} (see Note 1)	-0.3 V to 6 V
Positive output supply voltage range, V_{S+}	$V_{CC} - 0.3$ V to 15 V
Negative output supply voltage range, V_{S-}	-0.3 V to -15 V
Input voltage range, V_I : Driver	-0.3 V to $V_{CC} + 0.3$ V
Receiver	± 30 V
Output voltage range, V_O : T1OUT, T2OUT	$V_{S-} - 0.3$ V to $V_{S+} + 0.3$ V
R1OUT, R2OUT	-0.3 V to $V_{CC} + 0.3$ V
Short-circuit duration: T1OUT, T2OUT	Unlimited
Package thermal impedance, θ_{JA} (see Notes 2 and 3): D package	73°C/W
DW package	57°C/W
N package	67°C/W
NS package	64°C/W
Operating virtual junction temperature, T_J	150°C
Storage temperature range, T_{stg}	-65°C to 150°C

† Stresses beyond those listed under "absolute maximum ratings" may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated under "recommended operating conditions" is not implied. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.

NOTES: 1. All voltages are with respect to network GND.

2. Maximum power dissipation is a function of $T_J(\text{max})$, θ_{JA} , and T_A . The maximum allowable power dissipation at any allowable ambient temperature is $P_D = (T_J(\text{max}) - T_A) / \theta_{JA}$. Operating at the absolute maximum T_J of 150°C can affect reliability.
3. The package thermal impedance is calculated in accordance with JESD 51-7.

recommended operating conditions

		MIN	NOM	MAX	UNIT
V_{CC}	Supply voltage	4.5	5	5.5	V
V_{IH}	High-level input voltage (T1IN, T2IN)	2			V
V_{IL}	Low-level input voltage (T1IN, T2IN)			0.8	V
R1IN, R2IN	Receiver input voltage			± 30	V
T_A	Operating free-air temperature	MAX232	0	70	°C
		MAX232I	-40	85	

electrical characteristics over recommended ranges of supply voltage and operating free-air temperature (unless otherwise noted) (see Note 4 and Figure 4)

PARAMETER	TEST CONDITIONS	MIN	TYP†	MAX	UNIT
I_{CC} Supply current	$V_{CC} = 5.5$ V, All outputs open, $T_A = 25^\circ\text{C}$		8	10	mA

† All typical values are at $V_{CC} = 5$ V and $T_A = 25^\circ\text{C}$.

NOTE 4: Test conditions are C1-C4 = 1 μF at $V_{CC} = 5 \pm 0.5$ V.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MAX232, MAX232I

DUAL EIA-232 DRIVERS/RECEIVERS

SLLS047L - FEBRUARY 1989 - REVISED MARCH 2004

DRIVER SECTION

electrical characteristics over recommended ranges of supply voltage and operating free-air temperature range (see Note 4)

PARAMETER		TEST CONDITIONS	MIN	TYP†	MAX	UNIT
V _{OH}	High-level output voltage	T1OUT, T2OUT R _L = 3 kΩ to GND	5	7		V
V _{OL}	Low-level output voltage‡	T1OUT, T2OUT R _L = 3 kΩ to GND		-7	-5	V
r _o	Output resistance	T1OUT, T2OUT V _{S+} = V _{S-} = 0, V _O = ±2 V	300			Ω
I _{OS} §	Short-circuit output current	T1OUT, T2OUT V _{CC} = 5.5 V, V _O = 0		±10		mA
I _{IS}	Short-circuit input current	T1IN, T2IN V _I = 0			200	μA

† All typical values are at V_{CC} = 5 V, T_A = 25°C.

‡ The algebraic convention, in which the least-positive (most negative) value is designated minimum, is used in this data sheet for logic voltage levels only.

§ Not more than one output should be shorted at a time.

NOTE 4: Test conditions are C1-C4 = 1 μF at V_{CC} = 5 V ± 0.5 V.

switching characteristics, V_{CC} = 5 V, T_A = 25°C (see Note 4)

PARAMETER		TEST CONDITIONS	MIN	TYP	MAX	UNIT
SR	Driver slew rate	R _L = 3 kΩ to 7 kΩ, See Figure 2			30	V/μs
SR(t)	Driver transition region slew rate	See Figure 3		3		V/μs
	Data rate	One TOUT switching		120		kbit/s

NOTE 4: Test conditions are C1-C4 = 1 μF at V_{CC} = 5 V ± 0.5 V.

RECEIVER SECTION

electrical characteristics over recommended ranges of supply voltage and operating free-air temperature range (see Note 4)

PARAMETER		TEST CONDITIONS	MIN	TYP†	MAX	UNIT
V _{OH}	High-level output voltage	R1OUT, R2OUT I _{OH} = -1 mA	3.5			V
V _{OL}	Low-level output voltage‡	R1OUT, R2OUT I _{OL} = 3.2 mA			0.4	V
V _{IT+}	Receiver positive-going input threshold voltage	R1IN, R2IN V _{CC} = 5 V, T _A = 25°C		1.7	2.4	V
V _{IT-}	Receiver negative-going input threshold voltage	R1IN, R2IN V _{CC} = 5 V, T _A = 25°C	0.8	1.2		V
V _{hys}	Input hysteresis voltage	R1IN, R2IN V _{CC} = 5 V	0.2	0.5	1	V
r _i	Receiver input resistance	R1IN, R2IN V _{CC} = 5 V, T _A = 25°C	3	5	7	kΩ

† All typical values are at V_{CC} = 5 V, T_A = 25°C.

‡ The algebraic convention, in which the least-positive (most negative) value is designated minimum, is used in this data sheet for logic voltage levels only.

NOTE 4: Test conditions are C1-C4 = 1 μF at V_{CC} = 5 V ± 0.5 V.

switching characteristics, V_{CC} = 5 V, T_A = 25°C (see Note 4 and Figure 1)

PARAMETER		TYP	UNIT
t _{PLH(R)}	Receiver propagation delay time, low- to high-level output	500	ns
t _{PHL(R)}	Receiver propagation delay time, high- to low-level output	500	ns

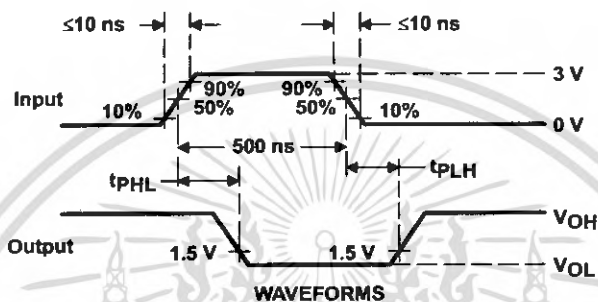
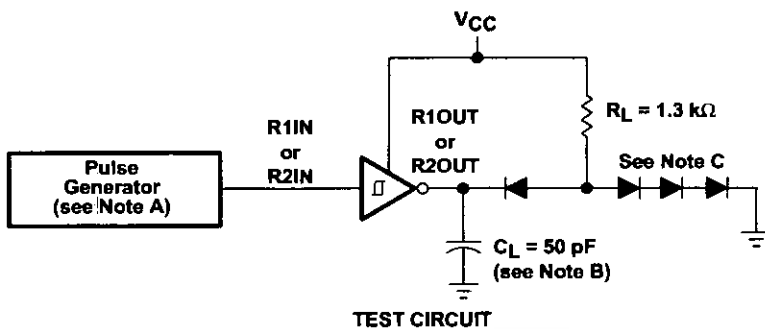
NOTE 4: Test conditions are C1-C4 = 1 μF at V_{CC} = 5 V ± 0.5 V.



POST OFFICE BOX 655303 • DALLAS, TEXAS 75265

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PARAMETER MEASUREMENT INFORMATION



- NOTES: A. The pulse generator has the following characteristics: $Z_O = 50 \Omega$, duty cycle $\leq 50\%$.
 B. C_L includes probe and jig capacitance.
 C. All diodes are 1N3064 or equivalent.

Figure 1. Receiver Test Circuit and Waveforms for t_{PHL} and t_{PLH} Measurements



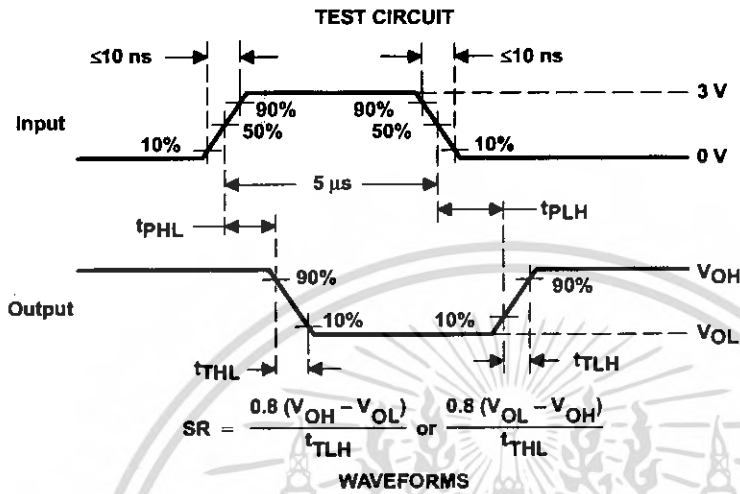
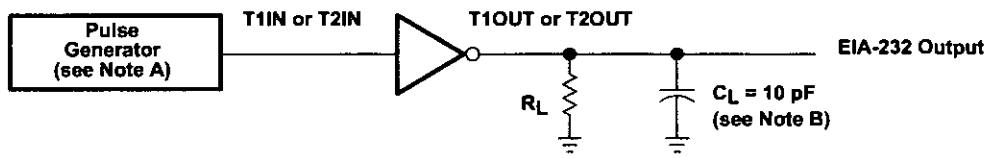
POST OFFICE BOX 855303 • DALLAS, TEXAS 75265

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MAX232, MAX232I DUAL EIA-232 DRIVERS/RECEIVERS

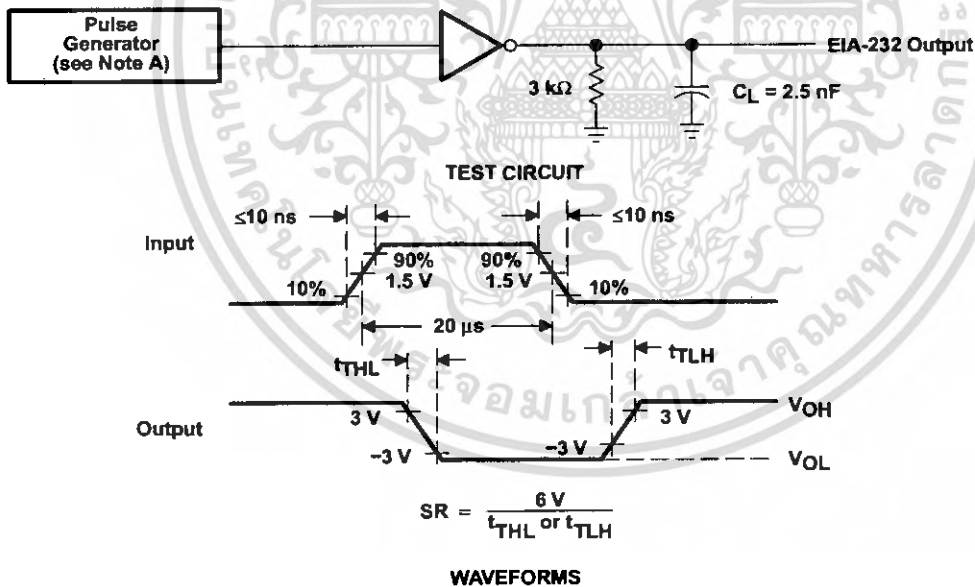
SLLS047L - FEBRUARY 1989 - REVISED MARCH 2004

PARAMETER MEASUREMENT INFORMATION



NOTES: A. The pulse generator has the following characteristics: $Z_O = 50 \Omega$, duty cycle $\leq 50\%$.
B. C_L includes probe and jig capacitance.

Figure 2. Driver Test Circuit and Waveforms for t_{PHL} and t_{PLH} Measurements (5- μ s Input)



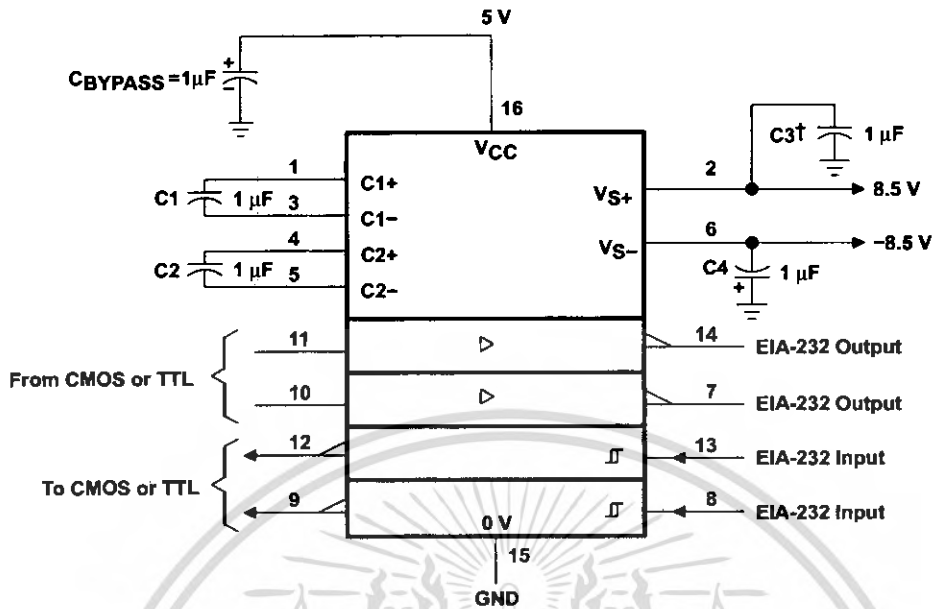
NOTE A: The pulse generator has the following characteristics: $Z_O = 50 \Omega$, duty cycle $\leq 50\%$.

Figure 3. Test Circuit and Waveforms for t_{THL} and t_{TLH} Measurements (20- μ s Input)

TEXAS
INSTRUMENTS

POST OFFICE BOX 855303 • DALLAS, TEXAS 75265

APPLICATION INFORMATION



† C3 can be connected to V_{CC} or GND.

NOTES: A. Resistor values shown are nominal.

B. Nonpolarized ceramic capacitors are acceptable. If polarized tantalum or electrolytic capacitors are used, they should be connected as shown. In addition to the 1-μF capacitors shown, the MAX202 can operate with 0.1-μF capacitors.

Figure 4. Typical Operating Circuit



POST OFFICE BOX 655303 • DALLAS, TEXAS 75285

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้