

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ระบบจำลองภาพ 3 มิติ

3 DIMENSION MODELLING SYSTEMS



2พ.
82442
2550

เลขหมู่.....
เลขทะเบียน..... 82442
วัน,เดือน,ปี..... 11 ก.ค. 2551

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาอิเล็กทรอนิกส์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2550

11๑๕๑๒๕๑
b.....
.....

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบจำลองภาพ 3 มิติ
3 DIMENSION MODELLING SYSTEMS

โดย
นางสาวชลธิชา พรหมเจริญ รหัส 47010149
นายชูธรรม ปิยะพงษ์ รหัส 47010188

อาจารย์ที่ปรึกษา
รศ.ดร. ชูชาติ ปิณฑวิรุจน์

ปริญญาานิพนธ์สำหรับปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาอิเล็กทรอนิกส์
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2550

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ ปีการศึกษา 2550

ภาควิชาอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบจำลองภาพ 3 มิติ(3 DIMENSION MODELLING SYSTEMS)

ผู้จัดทำ

1.นางสาวชลธิชา พรหมเจริญ

2.นายชูธรรม ปิยะพงษ์



อาจารย์ที่ปรึกษา

(รศ.ดร.ชูชาติ ปิณฑวิรุจน์)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบจำลองภาพ 3 มิติ

นางสาวชลธิชา พรหมเจริญ รหัส 47010149
นายชูธรรม ปิยะพงษ์ รหัส 47010188
รศ.ดร.ชูชาติ ปิณฑวิรุจน์ อาจารย์ที่ปรึกษา
ปีการศึกษา 2550

บทคัดย่อ

โครงการนี้นำเสนอการทำงานของระบบจำลองภาพแบบ 3 มิติจากวัตถุจริง มีหลักการทำงานคือ เมื่อกำลัง web-camera ซึ่งเชื่อมต่อกับคอมพิวเตอร์ด้วย USB port ทำหน้าที่รับภาพจากวัตถุที่วางบนแท่นหมุนสเต็ปมอเตอร์ ที่หมุนจนครบ 360 องศา โปรแกรมจะเก็บภาพที่ได้เป็นภาพนิ่งทีละภาพจนครบตามกำหนด แล้วนำภาพนิ่งทั้งหมดที่ได้รับไปประมวลผลโดยการใช้โปรแกรม C++Builder โดยมีกระบวนการทั้งหมดคือ การเก็บภาพ การสร้างภาพตัดขวางและการจำลองภาพ 3 มิติ ออกมาเป็น สำหรับแท่นหมุนที่รองรับวัตถุนั้นติดต่อกับคอมพิวเตอร์ผ่านทางพอร์ตอนุกรมควบคุมด้วยไมโครคอนโทรลเลอร์

3 DIMENSION MODELLING SYSTEM

Miss Chonticha Promchareon ID. 47010149

Mr. Chutum Piyapong ID. 47010188

Assist. Prof. Dr. Chuchat Pintavirooj Advisor

Educational Year 2007

Abstract

This project presents 3 dimension modelling system of object. The web-camera is interfaced with a computer a USB port. It takes image of the object. The object put on a platform is rotate 360 degree. C++Builder programming is used for processing all image and model 3 dimension of object. The program has 3 process which are acquisition, reconstruction and 3D modeling. A rotating base use platform stepping motor that interfaces with computer through a serial port and controlled by a microcontroller.

กิตติกรรมประกาศ

โครงการนี้สำเร็จลุล่วงไปด้วยดี ขอขอบพระคุณ รศ.ดร.ชูชาติ ปิณฑวิรุจน์ อาจารย์ที่ปรึกษา ที่กรุณาให้คำแนะนำและความคิดเห็นต่างๆ ในการทำงาน รวมทั้งเพื่อนๆ พี่ๆ ที่ห้องโปรเจ็คที่ให้ ความช่วยเหลือกันด้วยดีเสมอมา หากรายงานนี้มีข้อบกพร่องประการใด ขออภัยไว้ ณ ที่นี้

ชลธิชา พรหมเจริญ
ชูธรรม ปิยะพงษ์



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทกัณฑ์ย่อ	I
Abstract	II
กิตติกรรมประกาศ	III
สารบัญ	V
สารบัญรูปภาพ	VIII
สารบัญตาราง	IV
บทที่ 1 บทนำ	1
บทที่ 2 ทฤษฎี	3
2.1 ส่วนฮาร์ดแวร์	3
2.1.1 ไมโครคอนโทรลเลอร์	3
2.1.2 การสื่อสารแบบอนุกรม	9
2.1.3 ความรู้เบื้องต้นเกี่ยวกับมาตรฐาน RS-232	9
2.1.4 กิ่งและกิ่งการเชื่อมต่อของกิ่ง	10
2.1.5 สเต็ปมอเตอร์	12
2.1.6 วงจรขับ	15
2.2 ส่วนซอฟต์แวร์	18
2.2.1 การพัฒนาโปรแกรมด้วยภาษาซี	18
2.2.2 C++ และ C++Builder	19
2.2.3 การเขียนโปรแกรมแบบวิซวล	19
2.3 อัลกอริทึมของการสร้างภาพตัดขวางแบบกรวย	31
2.3.1 การสร้างภาพตัดขวางแบบพัด	32
2.3.2 การสร้างภาพตัดขวางแบบกรวย	39
2.3.3 โปรเจกชันของวัตถุ 3 มิติ	40
2.3.4 ฟิเตอร์เบ็คโปรเจกชันแบบ 3 มิติ	41
2.3.5 การแปลงเชิงเรขาคณิต	43
2.3.6 การสร้างภาพ 3 มิติโดยใช้ OpenGL	51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3 วงจรที่ใช้งานและการออกแบบโปรแกรม	56
3.1 วงจรที่ใช้งาน	56
3.1.1 ไอซี MAX 232	56
3.1.2 ไมโครคอนโทรลเลอร์	57
3.1.3 วงจรขับสเต็ปมอเตอร์	58
3.2 การออกแบบโปรแกรม	59
3.2.1 โปรแกรมในการควบคุมการหมุนของมอเตอร์	59
3.2.2 โปรแกรมในการควบคุมการทำงานของกล้อง	60
3.2.3 โปรแกรมในส่วนการประมวลผลภาพ	51
บทที่ 4 การทดลอง 52	
4.1 การทดลองส่วนของการควบคุมสเต็ปมอเตอร์	63
4.2 การทดลองส่วนการรับภาพและบันทึกภาพ	64
บทที่ 5 สรุปและวิเคราะห์ผลการดำเนินงาน	70
5.1 สรุปและวิเคราะห์ผลการทดลอง	70
5.2 ปัญหาในการดำเนินงาน	70
5.3 แนวทางในการพัฒนาต่อ	70
หนังสืออ้างอิง	
ภาคผนวก	

สารบัญรูปภาพ

	หน้า
บทที่ 1 บทนำ	
รูป 1.1 บล็อกไดอะแกรมการทำงานของระบบ	1
รูป 1.2 บล็อกไดอะแกรมการทำงานของการประมวลผลภาพ	2
บทที่ 2 ทฤษฎี	
รูป 2.1 ขาของ AT89C51	3
รูป 2.2 แสดงสัญญาณการรับและส่งข้อมูลในโหมด 0	8
รูป 2.3 แสดงสัญญาณการรับและส่งข้อมูลในโหมด 1	8
รูป 2.4 แสดงสัญญาณการรับและส่งข้อมูลในโหมด 2	8
รูป 2.5 แสดง DB-9 ที่ใช้ในการติดต่อผ่านพอร์ตอนุกรม	10
รูป 2.6 แสดง(ก) โครงสร้าง(ข) วงจรเทียบเท่าของมอเตอร์ชนิด 4 ขด	13
รูป 2.7 แสดงวิธีจับเอาท์พุทของสเต็ปมอเตอร์	16
รูป 2.8 วงจรสมมูลย์(equivalent circuit)ของสเต็ปมอเตอร์	16
รูป 2.9 การใช้ไดโอดชัฟเฟรสเซอร์	17
รูป 2.10 การใช้ไดโอดและตัวต้านทานชัฟเฟรสเซอร์	17
รูป 2.11 การใช้ซีเนอร์ไดโอดชัฟเฟรสเซอร์	17
รูป 2.12 การใช้ตัวเก็บประจุชัฟเฟรสเซอร์	18
รูป 2.13 ฟอรัมที่ใช้ในการเขียนโปรแกรม	21
รูป 2.14 Unit.ccp	25
รูป 2.15 Unit.h	26
รูป 2.16 component กลุ่ม standard	27
รูป 2.17 component กลุ่ม Additional	27
รูป 2.18 component win32	27
รูป 2.19 component กลุ่ม system	27
รูป 2.20 component กลุ่ม data access	27
รูป 2.21 component กลุ่ม data control	27
รูป 2.22 component กลุ่ม dbExpress	28
รูป 2.23 component กลุ่ม datasnap	28

รูป 2.24 component กลุ่ม BDE	28
รูป 2.25 component กลุ่ม AD0	28
รูป 2.26 component กลุ่ม InterBase	28
รูป 2.27 component กลุ่ม Webservice	28
รูป 2.28 component กลุ่ม InternetExpress	28
รูป 2.29 component กลุ่มInternet	28
รูป 2.30 component กลุ่ม WebSnap	29
รูป 2.31 component กลุ่ม FastNet	29
รูป 2.32 component กลุ่ม Decision Cube	29
รูป 2.33 component กลุ่ม QReport	29
รูป 2.34 component กลุ่ม Dialog	29
รูป 2.35 component กลุ่ม Win3.1	29
รูป 2.36 component กลุ่ม Samples	29
รูป 2.37 component กลุ่ม ActiveX	30
รูป 2.38 component กลุ่ม COM+	30
รูป 2.39 component กลุ่ม Interbase Admin	30
รูป 2.40 component กลุ่ม Servers	30
รูป 2.41 component กลุ่ม Indy Cilent	30
รูป 2.42 component กลุ่ม Indy Servers	30
รูป 2.43 component กลุ่ม Indy Misc	31
รูป 2.44 component กลุ่ม Office2k	31
รูป 2.45 เส้นทางเดินแสง 32	
รูป 2.46 ข้อมูล โปรเจคชันเมื่อพิจารณาทางเดินแสงแบบพัดที่มีระยะห่างระหว่างหัวรับ	
เท่ากัน	33
รูป 2.47 ข้อมูล โปรเจคชันเมื่อพิจารณาทางเดินแสงแบบพัดที่มีระยะห่างระหว่างหัวรับ	
เท่ากัน	34
รูป 2.48 ข้อมูล โปรเจคชันเมื่อพิจารณาทางเดินแสงแบบพัดที่มีระยะห่างระหว่างหัวรับ	
เท่ากัน	37
รูป 2.49 เรขาคณิตของทางเดินแสงแบบพัดที่พิจารณาระยะห่างระหว่างหัวรับเท่ากัน	37
รูป 2.50 การสร้างภาพตัดขวางจากวัตถุ 3 มิติ	39
เอกสารรูป 2.53 การพิจารณาทางเดินแสงแบบกรวยในรูปแบบทางเดินแสงแบบพัดนำไปใช้ประโยชน์ด้านการค้า	39
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้	

รูป 2.54 ความสัมพันธ์ของการหมุนของระบบ โคออร์ดิเนต	40
รูป 2.55 ภาพแสดงระบบพิกัด (t,s)	42
รูป 2.56 ส่วนประกอบของระบบพิกัด โฮโมจีเนียส	44
รูป 2.57 การสเกลวัตถุ	46
รูป 2.58 การแปลงแบบหมุนรอบแกนทั้งสาม	47
รูป 2.59 การสะท้อนวัตถุกับระนาบ XY	49
รูป 2.60 การเลื่อน โดยสัมพันธ์กับค่า Z	50
รูป 2.61 รูปเหลี่ยมที่เหมาะสมและไม่เหมาะสม	52
รูป 2.62 รูปเหลี่ยมที่ไม่ได้ระนาบเมื่อหมุนเปลี่ยนมุมมองจะทำให้เกิดส่วนเว้า	52
รูป 2.63 เส้นโค้งที่เกิดจากการประกอบกันของเส้นตรง	53
รูป 2.64 ทรงกลมที่เกิดจากการประกอบกันของพื้นผิวเรียบ	53
รูป 2.65 ทรงกลมที่ไม่แสดงตกระทบและทรงกลมที่มีแสดงตกระทบ	53
บทที่ 3 วงจรที่ใช้งานและการออกแบบโปรแกรม	
รูป 3.1 แสดงขาของไอซี MAX232	56
รูป 3.2 แสดงขาไอซี AT89C51	57
รูป 3.3 วงจรที่ใช้ส่งข้อมูลจากคอมพิวเตอร์ไปยังไมโครคอนโทรลเลอร์	58
รูป 3.4 วงจรที่ใช้ในการขับสเต็ปมอเตอร์	58
รูป 3.5 แผนภาพการทำงานของมอเตอร์ที่รับค่าจากพอร์ตอนุกรม	59
รูป 3.6 บล็อกไดอะแกรมแสดงการทำงานของโปรแกรมส่วนการเก็บภาพ	60
รูป 3.7 บล็อกไดอะแกรมแสดงการทำงานของโปรแกรมส่วนการสร้างภาพตัดขวาง	61
รูป 3.8 บล็อกไดอะแกรมแสดงการทำงานของโปรแกรมส่วนการสร้างภาพ 3 มิติ	62
บทที่ 4 การทดลอง	
รูป 4.1 โปรแกรมที่ออกแบบ	64
รูป 4.2 แสดงการรับภาพจากกล้องแสดงบนโปรแกรม	65
รูป 4.3 แสดงข้อมูลภาพที่ได้และบันทึกใน Folder ที่กำหนด	66
รูป 4.4 Dialog box ปรับแต่งค่าของกล้อง	66
รูป 4.5 แสดงโปรแกรมส่วนของการสร้างภาพตัดขวาง	67
รูป 4.6 แสดงการใช้งานโปรแกรมในส่วนของการสร้างภาพตัดขวาง	67
รูป 4.7 แสดงภาพตัดขวางที่ได้จากการใช้โปรแกรม	68
รูป 4.8 แสดงโปรแกรมส่วนของการสร้างแบบจำลอง 3 มิติ	68
เอกสารรูป 4.9 แสดงแบบจำลอง 3 มิติที่ได้จากโปรแกรมศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประ ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้	69

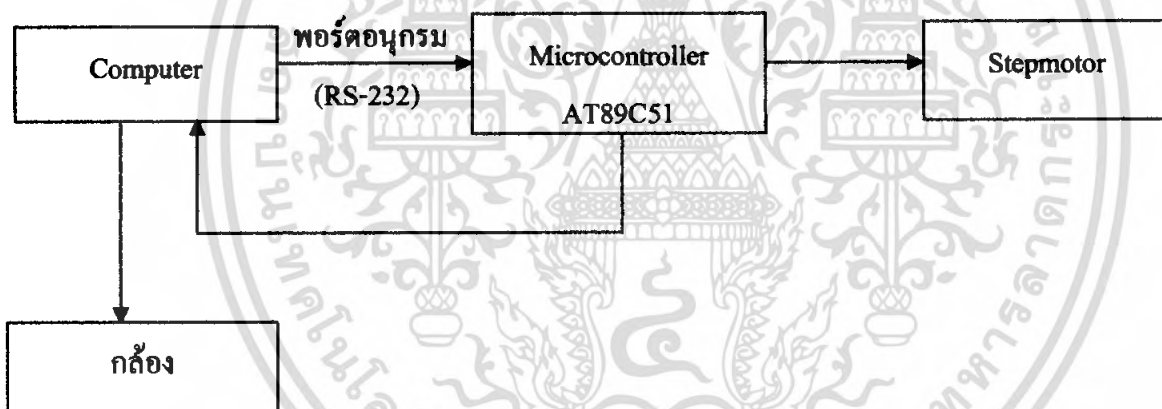
สารบัญตาราง

	หน้า
บทที่ 2 ทฤษฎี	
ตารางที่ 2.1 แสดงคุณสมบัติพิเศษของพอร์ต 3 ของ AT89C51	4
ตารางที่ 2.2 แสดงบิตต่างๆ ของ SCON	6
ตารางที่ 2.3 แสดงโหมดการทำงานของพอร์ตอนุกรม	6
ตารางที่ 2.4 แสดงหน้าที่ต่างๆ ของขา DB-9	10
ตารางที่ 2.5 แสดงมุมของ โรเตอร์เทียบกับกระแสไฟฟ้าที่จ่ายแก่เฟสต่างๆ 8 ตำแหน่ง	13
ตารางที่ 2.6 แสดงการควบคุมการหมุนของมอเตอร์แบบเวฟ	14
ตารางที่ 2.7 แสดงการควบคุมการหมุนของมอเตอร์แบบ Full step	14
ตารางที่ 2.8 แสดงการควบคุมการหมุนของมอเตอร์แบบ Halfstep	15
บทที่ 3 วงจรที่ใช้งานและการออกแบบโปรแกรม	
ตารางที่ 3.1 ตารางแสดงหน้าที่การทำงานของขาต่างๆ ของ MAX232	56
ตารางที่ 3.2 ตารางแสดงคุณสมบัติพิเศษของพอร์ต 3 ของ AT89C51	57
บทที่ 4 การทดลอง	
ตารางที่ 4.1 ขนาดมุมของมอเตอร์และจำนวนภาพที่ได้	63
ตารางที่ 4.2 แสดงค่าของมุมที่กำหนดและมุมที่ได้จริง	63
ตารางที่ 4.3 แสดงความแม่นยำของมุมเมื่อมีการทำซ้ำ	64

บทที่ 1

บทนำ

โครงการนี้กล่าวถึงการจำลองภาพ 3 มิติจากวัตถุ โดยทำการรับภาพนิ่งและภาพวิดีโอจากกล้องเว็บแคมฯ ในมุมต่างๆ ที่ได้โดยการหมุนวัตถุให้กล้องจับภาพจนครบ 360 องศา จากนั้นนำภาพที่ได้รับมาประมวลผลภาพโดยใช้โปรแกรม ซี-พลัส-พลัส บิลเดอร์(C++Builder) เพื่อจำลองออกมาเป็นภาพ 3 มิติปรากฏบนโปรแกรมที่ออกแบบ สำหรับการหมุนของวัตถุนั้นควบคุมด้วยคอมพิวเตอร์ส่งคำสั่งผ่านทางพอร์ตอนุกรมไปยังไมโครคอนโทรลเลอร์(Microcontroller) AT89C51 เพื่อควบคุมการทำงานของแท่นหมุน แล้วไมโครคอนโทรลเลอร์จึงส่งสัญญาณกลับมาที่คอมพิวเตอร์เพื่อให้โปรแกรมจับภาพ จากนั้นจะนำภาพที่บันทึกไว้เข้าสู่กระบวนการต่อไปคือการสร้างภาพตัดขวาง และจำลองภาพ 3 มิติจากภาพตัดขวางที่ได้

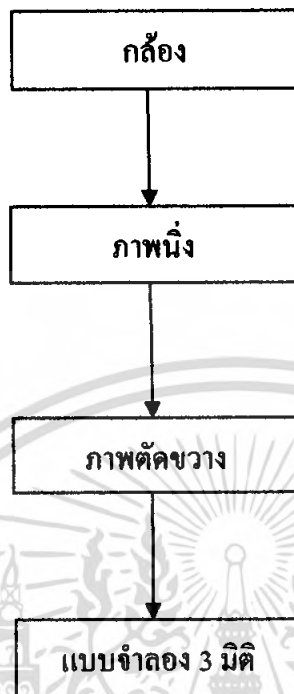


รูป 1.1 บล็อกไดอะแกรมการทำงานของระบบ

จากรูปที่ 1.1 ประกอบไปด้วยส่วนต่างๆ ดังต่อไปนี้

- 1.กล้องเว็บแคมฯ เป็นตัวจับภาพและส่งภาพที่จับได้ไปยังคอมพิวเตอร์ผ่าน ยูเอสบี พอร์ต (USB Port) เพื่อประมวลผลต่อไป
- 2.คอมพิวเตอร์ จะเป็นตัวทำการประมวลผลภาพที่ได้จากกล้อง โดยใช้โปรแกรม C++Builder และยังเป็นตัวส่งคำสั่งไปให้ไมโครคอนโทรลเลอร์และรับสัญญาณจากไมโครคอนโทรลเลอร์ด้วย
- 3.ไมโครคอนโทรลเลอร์ เบอร์ AT89C51 เป็นตัวรับคำสั่งจากเครื่องคอมพิวเตอร์ เพื่อควบคุมการทำงานของสเต็ปมอเตอร์ และส่งสัญญาณกลับไปยังคอมพิวเตอร์
- 4.สเต็ปมอเตอร์ ใช้ในการหมุนแท่นรองรับวัตถุเพื่อให้กล้องจับภาพในมุมต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 1.2 บล็อกไดอะแกรมการทำงานของกระบวนการประมวลผลภาพ

วัตถุประสงค์

1. เพื่อศึกษาการนำภาพที่ได้รับจากวัตถุจริงมาประมวลผลภาพและจำลองออกมาเป็น 3 มิติ

ขอบเขตของโครงการ

1. แทนรองรับวัตถุสามารถหมุนได้โดยการสั่งจากโปรแกรม ซี-พลัส-พลัส บิลเคอร์ผ่านทางพอร์ตอนุกรมไปยังคอนโทรลเลอร์
2. โปรแกรมสามารถรับและบันทึกภาพผ่านกล้องได้
3. โปรแกรมสามารถสามารถนำภาพที่บันทึกมาประมวลผลเป็นภาพจำลอง 3 มิติได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

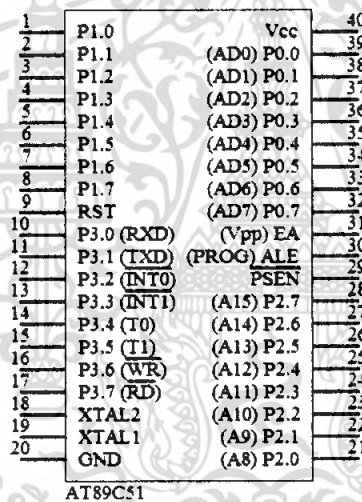
บทที่ 2

ทฤษฎี

2.1 ส่วนฮาร์ดแวร์

2.1.1 ไมโครคอนโทรลเลอร์

ไมโครคอนโทรลเลอร์ นั้นประกอบไปด้วย หน่วยประมวลผล หน่วยคำนวณทางคณิตศาสตร์และลอจิก วงจรรับสัญญาณอินพุต วงจรขับสัญญาณทางเอาต์พุต หน่วยความจำ และ วงจรกำเนิดสัญญาณนาฬิกา ซึ่งสามารถนำไปประยุกต์ใช้งานทางอิเล็กทรอนิกส์ได้หลายด้าน โดยที่สามารถเลือกจากความต้องการสร้างระบบควบคุมขนาด 8 บิต มีการเชื่อมต่ออุปกรณ์ภายนอกไม่มากนัก โดยในโครงการนี้ใช้ไมโครคอนโทรลเลอร์เบอร์ AT89C51 ซึ่งมีขาแสดงดังรูป



1	P1.0	Vcc	40
2	P1.1	(AD0) P0.0	39
3	P1.2	(AD1) P0.1	38
4	P1.3	(AD2) P0.2	37
5	P1.4	(AD3) P0.3	36
6	P1.5	(AD4) P0.4	35
7	P1.6	(AD5) P0.5	34
8	P1.7	(AD6) P0.6	33
9	RST	(AD7) P0.7	32
10	P3.0 (RXD)	(Vpp) EA	31
11	P3.1 (TXD)	(PROG) ALE	30
12	P3.2 (INT0)	PSEN	29
13	P3.3 (INT1)	(A15) P2.7	28
14	P3.4 (T0)	(A14) P2.6	27
15	P3.5 (T1)	(A13) P2.5	26
16	P3.6 (WR)	(A12) P2.4	25
17	P3.7 (RD)	(A11) P2.3	24
18	XTAL2	(A10) P2.2	23
19	XTAL1	(A9) P2.1	22
20	GND	(A8) P2.0	21

AT89C51

รูป 2.1 ขาของ AT89C51

ไมโครคอนโทรลเลอร์ตระกูล MCS-51 มีจำนวนมาก ขึ้นกับโครงสร้างภายใน บางเบอร์มีหน่วยความจำภายในเป็นแบบรอม บางเบอร์เป็นแบบอีพรอม บางเบอร์มีแรมภายใน 128 ไบต์ บางเบอร์มี 256 ไบต์ เป็นต้น คุณสมบัติที่สำคัญของ MCS-51 มีดังนี้

- หน่วยความจำรอม 4 กิโลไบต์
- หน่วยความจำแรม 128 ไบต์
- พอร์ตอินพุตและเอาต์พุต ขนาด 8 บิต 4 พอร์ต
- ไทเมอร์ 16 บิต 2 ตัว
- สามารถอินเทอร์รัพท์ได้ 5 แหล่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- วงจรออสซิลเลเตอร์และวงจรมานานาฬิกานานาฬิกา
- พอร์ตอนุกรมสามารถรับส่งข้อมูลแบบ ฟูล ดุเพล็กซ์ ความเร็วสูง
- ี้าหน่วยความจำโปรแกรมภายนอกได้ 64 กิโลไบต์
- ี้าหน่วยความจำข้อมูลภายนอกได้ 64 กิโลไบต์
- สามารถประมวลผลทีละบิต
- ี้าหน่วยความจำแบบบิตได้ 210 ตำแหน่ง
- หนึ่งวัฏจักรคำสั่งกินเวลาประมาณ 1 ไมโครวินาทีขณะทำงานด้วย Clock 12 เมกะเฮิร์ต

ความหมายต่างๆ มีดังนี้

1. พอร์ต 0 (Port 0)

พอร์ต 0 ได้แก่ขา 32-39 สามารถเป็นพอร์ตอินพุตและเอาต์พุตได้ ซึ่งในการติดต่อกับหน่วยความจำภายนอกสามารถเป็นขาบัสดำแหน่ง และบัสดข้อมูลได้

2. พอร์ต 1 (Port 1)

พอร์ต 1 ได้แก่ขา 1-8 เป็นพอร์ต 8 บิต สามารถอ้างทีละบิตได้ เป็น P1.0, P1.1,...etc

3. พอร์ต 2 (Port 2)

พอร์ต 2 ได้แก่ขา 21-28 สามารถทำงานได้ 2 หน้าที เป็นพอร์ต 8 บิต และเป็นขาแอดเดรส 8 บิต ในการอ้างหน่วยความจำภายนอก

4. พอร์ต 3 (Port 3)

พอร์ต 3 ได้แก่ขา 10-17 สามารถทำงานได้ 2 หน้าที เป็นพอร์ตอินพุตและเอาต์พุต และเป็นขาควบคุมต่างๆ ดังตารางที่ 2.1

ตารางที่ 2.1 แสดงคุณสมบัติพิเศษของพอร์ต 3 ของ AT89C51

บิต	ชื่อ	หน้าที่พิเศษ
P3.0	RXD	ใช้รับข้อมูลทางพอร์ตอนุกรม
P3.1	TXD	ใช้ส่งข้อมูลทางพอร์ตอนุกรม
P3.2	$\overline{\text{INT0}}$	อินเทอร์รัพท์ภายนอกหมายเลข 0
P3.3	$\overline{\text{INT1}}$	อินเทอร์รัพท์ภายนอกหมายเลข 1
P3.4	T0	ตัวจับเวลา/ตัวนับ ตัวที่ 0
P3.5	T1	ตัวจับเวลา/ตัวนับ ตัวที่ 1
P3.6	$\overline{\text{WR}}$	สัญญาณเขียนข้อมูลหน่วยความจำภายนอก
P3.7	$\overline{\text{RD}}$	สัญญาณอ่านข้อมูลหน่วยความจำภายนอก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. $\overline{\text{PSEN}}$ (Program Store Enable)

ขา $\overline{\text{PSEN}}$ ได้แก่ขา 29 เป็นขาที่ส่งสัญญาณออก ขานี้ทำการแอกทีฟเมื่อ MCS-51 ต้องการอ่าน Code โปรแกรมภายนอก โดยปกติถ้าหน่วยความจำภายนอกเป็นอีพรอมขา $\overline{\text{PSEN}}$ ต่อกับขา Output Enable $\overline{\text{OE}}$ ของอีพรอม

6. ALE (Address Latch Enable)

เนื่องจากพอร์ท 0 สามารถเป็นขาอ้างตำแหน่ง และขาข้อมูล MCS-51 ซึ่งมีขา ALE ได้แก่ขา 30 ขานี้ใช้ผสมสัญญาณบัสตำแหน่งของพอร์ท 0 ในการใช้งานระบบ MCS-51 นั้น ต้องมีอุปกรณ์มาต่อกับพอร์ท 0 ที่ทำหน้าที่ Latch สัญญาณบัสตำแหน่ง เมื่อ MCS-51 ติดต่อกับ หน่วยความจำภายนอก MCS-51 ส่งสัญญาณบัสตำแหน่งออกมาก่อนทาง พอร์ท 0 จากนั้นส่งสัญญาณ ALE มา Latch อุปกรณ์ภายนอก ให้เก็บค่าบัสตำแหน่งของพอร์ท 0 ไว้เพื่อใช้พอร์ท 0 เป็นบัสข้อมูลต่อไป

7. $\overline{\text{EA}}$ (External Access)

ขา $\overline{\text{EA}}$ ได้แก่ขา 31 ในสถานะเป็นลอจิก "1" ใช้กับเบอร์ 8051/8052 เพื่อให้อ่านโปรแกรมจากหน่วยความจำภายใน และในสถานะเป็นลอจิก "0" ทำให้ MCS-51 ทำโปรแกรมโดยอ่านจากหน่วยความจำโปรแกรมภายนอก (เมื่อ $\overline{\text{EA}}$ เป็น "0" ขา $\overline{\text{PSEN}}$ ทำการแอกทีฟ) หากเป็นเบอร์ 8031 หรือ 8032 ขา $\overline{\text{EA}}$ เป็น "0" เสมอ เพราะไม่มีโปรแกรมหน่วยความจำภายใน หากใช้เบอร์ 8051/8052 ซึ่งมีหน่วยความจำภายในและให้ขา $\overline{\text{EA}}$ เป็น "0" ซึ่งหยุดการทำงานของรอมภายในและอ่านโปรแกรมจากอีพรอมภายนอกแทน

8. RST (Reset)

ขา RST ได้แก่ขา 9 เป็นขารีเซ็ต MCS-51 โดยให้เป็นสถานะลอจิก "1" อย่างน้อย 2 Machine Cycles จึงสามารถรีเซ็ตระบบได้

การรับส่งข้อมูลผ่านพอร์ทอนุกรมของไมโครคอนโทรลเลอร์

ไมโครคอนโทรลเลอร์ตระกูล MCS-51 จะมีวงจรสื่อสารอนุกรมแบบฟูลดูเพล็กซ์(วงจรสื่อสารแบบฟูลดูเพล็กซ์ หมายถึง วงจรที่สามารถทำการรับและส่งข้อมูลในลักษณะ 2 ทิศทางได้ในเวลาเดียวกัน) โดยใช้ขาสัญญาณของพอร์ท 3 คือขา P3.0 เป็นขา รับข้อมูลเข้าหรือ Rx D และขา P3.1 เป็นขา ส่งข้อมูลออกหรือ Tx D โดยวงจรสื่อสารข้อมูลแบบอนุกรมของไมโครคอนโทรลเลอร์ตระกูล MCS-51 แบบแฟลช เป็นแบบอะซิงโครนัส

รีจิสเตอร์ที่เกี่ยวข้องกับการทำงานของพอร์ตอนุกรมใน MCS-51

ในการทำงานของพอร์ตอนุกรมไมโครคอนโทรลเลอร์ MCS-51 มีรีจิสเตอร์ที่เกี่ยวข้องอยู่ 2 ตัวดังนี้

รีจิสเตอร์บัฟเฟอร์ของพอร์ตอนุกรมหรือ SBUF (Serial data buffer register) มีขนาด 8 บิต แบ่งเป็น 2 ส่วนคือ รีจิสเตอร์บัฟเฟอร์สำหรับส่งข้อมูล(Transmit buffer register) และรับข้อมูล (receive buffer register) เมื่อมีการเขียนข้อมูลมายังรีจิสเตอร์บัฟเฟอร์ของพอร์ตอนุกรม ข้อมูลนั้นจะถูกส่งออกไปยังบัฟเฟอร์สำหรับส่งข้อมูล เพื่อส่งจากไมโครคอนโทรลเลอร์ผ่านทางขา TxD หรือ P3.1 ในกรณีที่มีการอ่านข้อมูลจากรีจิสเตอร์บัฟเฟอร์ของพอร์ตอนุกรม ข้อมูลจะถูกส่งผ่านไปยังรีจิสเตอร์บัฟเฟอร์สำหรับรับข้อมูลเพื่อส่งต่อไปยังไมโครคอนโทรลเลอร์ต่อไป สำหรับการรับข้อมูลอนุกรมจากภายนอกนั้นจะผ่านมาทางขา RxD หรือขา P3.0 ของไมโครคอนโทรลเลอร์ MCS-51

รีจิสเตอร์ควบคุมการทำงานของพอร์ตอนุกรมหรือ SCON (Serial port control register) SCON เป็นรีจิสเตอร์ที่ใช้ควบคุมการทำงานของพอร์ตอนุกรม ซึ่งประกอบด้วยบิตต่างๆ 8 บิตดังนี้

ตารางที่ 2.2 แสดงบิตต่างๆ ของ SCON

บิต 7	บิต 6	บิต 5	บิต 4	บิต 3	บิต 2	บิต 1	บิต 0
SM0	SM1	SM2	REN	TB8	RB8	TI	RI

SM0 (Serial port mode bit 0): ปกติจะใช้ร่วมกับบิต SM1 เพื่อกำหนดโหมดการทำงานของพอร์ตอนุกรม การเข้าถึงบิตนี้จะเกิดขึ้นได้ก็ต่อเมื่อมีการเคลียร์บิต SMOD ซึ่งก็คือ บิต 6 ของรีจิสเตอร์ PCON ในกรณีที่ใช้ความสามารถในการตรวจจับข้อผิดพลาดของเฟรมบิตข้อมูลนี้ บิตนี้จะแจ้งความผิดพลาดโดยจะเซตเป็น 1 ทันที เมื่อพบว่าไม่สามารถตรวจจับบิตหยุด(stop bit) ของข้อมูลพอร์ตอนุกรมได้

SM1 (serial port mode bit 1): ใช้ร่วมกับบิต SM0 ในการกำหนดโหมดการทำงานของพอร์ตอนุกรม โดยกำหนดได้ดังนี้

ตารางที่ 2.3 แสดงโหมดการทำงานของพอร์ตอนุกรม

SM0	SM1	โหมด	รายละเอียด	อัตราบอด(Buad rate)
0	0	0	ซีพรีจิสเตอร์	ความถี่สัญญาณนาฬิกา/6
0	1	1	UART 8 บิต	ปรับค่าได้
1	0	2	UART 9 บิต	ความถี่สัญญาณนาฬิกา/2
1	1	3	UART 9 บิต	ปรับค่าได้

SM2 (serial port mode bit 1): ใช้ในการเอ็นเนเบิลการสื่อสารในแบบมัลติโพรเซสเซอร์ (multiprocessor) ในการทำงานของโหมด 2 และ 3 ของพอร์ตอนุกรมในไมโครคอนโทรลเลอร์ MCS-51 ถ้าบิตนี้เป็น "1" บิต RI จะไม่แอกทีฟ ถ้าบิตที่ 9 ที่เข้ามาเป็น "0" (ข้อมูลบิตที่ 9 เก็บไว้ที่ บิต RB8) ในการทำงานโหมด 1 ถ้าบิตนี้เซต บิต RI จะไม่แอกทีฟถ้ายังไม่ได้รับบิตหยุด ส่วนใน โหมด 0 บิตนี้ไม่มีการใช้งาน

REN (Enable serial reception): ใช้ในการเอ็นเนเบิลความสามารถในการรับข้อมูลของพอร์ตอนุกรม "1" เอ็นเนเบิลการรับข้อมูล และ "0" ดิสเอเบิลการรับข้อมูล

TB8 (Transmit data bit 8): ใช้สำหรับเก็บข้อมูลบิตที่ 9 ที่ต้องการส่งออกไปในการทำงาน โหมด 2 และ 3 ของพอร์ตอนุกรม

RB8 (Receive data bit 8): ใช้เก็บข้อมูลบิต 8 ของข้อมูลที่รับเข้ามาของพอร์ตอนุกรมเมื่อทำงานในโหมด 2 และ 3 ในโหมด 1 ถ้าบิต SM2 = 0 ข้อมูลของบิตหยุดจะเก็บไว้ที่บิตนี้ ไม่ใช้งาน บิตนี้ในโหมด

TI (Transmit Interrupt Flag): ใช้ในการแสดงการเกิดอินเตอร์รัพท์เมื่อมีการส่งข้อมูลออกจากพอร์ตอนุกรม เมื่อทำงานใน โหมด 0 บิตนี้จะเซตเมื่อมีการส่งข้อมูลบิต 7 หรือ บิตที่ 8 ออกไป แต่ ถ้าทำงานในโหมดอื่นบิตนี้จะเซตเมื่อมีการเริ่มต้นส่งบิตหยุดหรือบิตปิดท้าย

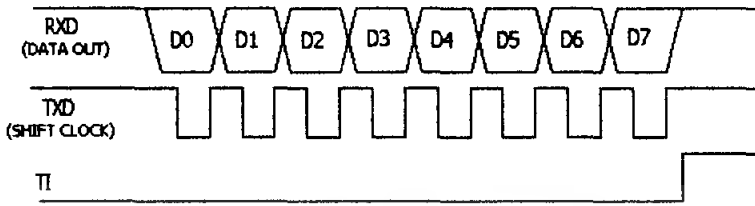
RI (Receive Interrupt flag): ใช้ในการเกิดอินเตอร์รัพท์เมื่อมีการรับข้อมูลเข้าสู่พอร์ตอนุกรม เมื่อทำงานใน โหมด 0 บิตนี้จะเซตเมื่อมีการส่งข้อมูลบิต 7 หรือบิตที่ 8 เสร็จสมบูรณ์ แต่ถ้าทำงานใน โหมดอื่น บิตนี้จะเซตเมื่อการรับบิตปิดท้ายดำเนินไปได้ครึ่งทาง นอกจากนี้การเซตบิตนี้ยังมีเงื่อนไขที่กำหนด โดยบิต SM2 ร่วมด้วย

โหมดการติดต่อทางพอร์ตอนุกรมของไมโครคอนโทรลเลอร์

การสื่อสารอนุกรมของไมโครคอนโทรลเลอร์จะแบ่งออกได้เป็น 4 โหมดด้วยกัน และในแต่ละโหมดจะสามารถสรุปหน้าที่ได้ดังนี้

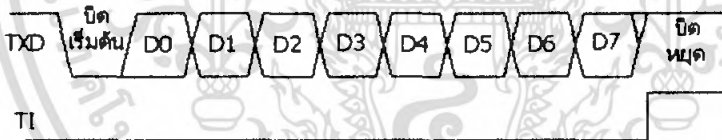
1. โหมด 0 : จะเป็นการรับส่งข้อมูลขนาด 8 บิตแบบอนุกรม การส่งข้อมูลจะเลื่อนออกไปทีละบิต โดยจะใช้งานของขา RxD เพียงขาเดียว และจะไม่มีการส่งบิตเริ่มต้น (Start bit) ส่วนขา TxD จะใช้เป็นขาของสัญญาณนาฬิกาในการให้จังหวะการเลื่อน ข้อมูลกับวงจรถ่ายนอก (Shift clock) อัตราการรับส่งข้อมูล (Baud rate) จะเป็น 1/12 เท่าของสัญญาณนาฬิกา การรับและส่งข้อมูลจะเริ่มจากบิตต่ำ (LSB) ก่อน ใช้สำหรับเป็นชิพทรีจิสเตอร์ (Shift Register) จุดประสงค์เพื่อใช้ในการขยายพอร์ตอินพุต หรือพอร์ตเอาต์พุต ให้มีจำนวนมาก แต่ในโหมด 0 เรามักจะไม่ค่อยนิยมนำมาใช้งาน

เพราะไอซี ไมโครคอนโทรลเลอร์ในปัจจุบันที่เราใช้อยู่ นั้น มีจำนวนพอร์ตที่มากพอ และมีไอซีเบอร์อื่นๆ ในตระกูลเดียวกันให้เลือกจำนวนพอร์ตใช้งานมากมายอยู่แล้ว



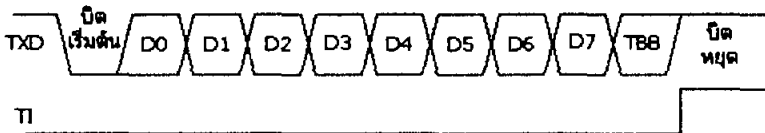
รูป 2.2 แสดงสัญญาณการรับและส่งข้อมูลใน โหมด 0

2.โหมด 1 : จะเป็นการรับและส่งข้อมูลขนาด 10 บิตแบบ UART (Universal Asynchronous Receiver Transmitters) สามารถใช้ในการติดต่อสื่อสารอนุกรมกับมาตรฐานของ RS-232C ของไมโครคอมพิวเตอร์ได้ ซึ่งข้อมูลอนุกรม 10 บิต จะเข้ามาทางขา RXD และ ส่งข้อมูลออกแบบอนุกรมทางขา TXD โดยจะประกอบด้วย 1 บิตแรกเป็นบิตเริ่มต้น (Start bit ค่า 0) , 8 บิตต่อมาจะเป็นบิตของข้อมูล (การรับ/ส่งจะเริ่มจากบิตต่ำก่อน) และ บิตหยุดอีก 1 บิต (Stop bit ค่า 1) ส่วนทางด้านรับข้อมูลจะนำค่าบิตหยุด (Stop bit) ที่รับเข้ามาได้นำไปเก็บไว้ในบิต RB8 ที่อยู่ในรีจิสเตอร์ SCON และความเร็วของการส่งข้อมูลใน โหมด 1 จะขึ้นอยู่กับบิต SMOD ที่อยู่ในรีจิสเตอร์ PCON และอัตราโอเวอร์โฟลว์ของไทมเมอร์ 1 ซึ่งอัตราการรับส่งข้อมูลใน โหมดนี้สามารถกำหนดได้ตามต้องการ



รูป 2.3 แสดงสัญญาณการส่งข้อมูลใน โหมด 1

3.โหมด 2 : จะเป็นการรับและส่งข้อมูลขนาด 11 บิตแบบ UART (Universal Asynchronous Receiver Transmitters) ข้อมูลแบบอนุกรมจะถูกรับเข้ามาทางขา RXD และส่งข้อมูลออกไปทางขา TXD ซึ่งข้อมูล 11 บิตประกอบด้วย บิตแรกจะเป็นบิตเริ่มต้น (Start bit ค่า 0) , 9 บิตต่อมาจะเป็นบิตของข้อมูล และบิตสุดท้ายจะเป็นบิตหยุด 1 บิต (Stop bit ค่า 1) สำหรับข้อมูลในบิตที่ 9 จะกำหนดไว้ใน TB8 ที่อยู่ในรีจิสเตอร์ SCON ซึ่งสามารถกำหนดเป็น 1 หรือ 0 ก็ได้ นิยมนำมาใช้ในการส่งบิตเพื่อตรวจสอบการส่งข้อมูล (Parity bit)



รูป 2.4 แสดงสัญญาณการส่งข้อมูลใน โหมด 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. โหมด 3 : เป็นการรับส่งข้อมูลแบบ 11 บิตแบบ UART (Universal Asynchronous Receiver Transmitters) เหมือนกับโหมด 2 แต่ในโหมด 3 สามารถกำหนดอัตราความเร็วในการรับและส่งข้อมูลได้ตามต้องการ

UART (Universal Asynchronous Receiver Transmitters) : เป็นการส่งข้อมูลแบบอนุกรม โดยขึ้นอยู่กับความพร้อมของทางด้านส่ง และด้านรับ เป็นการส่งข้อมูลโดยทำการเพิ่มเติมข้อมูลบางอย่างเข้าไป (Start bit ,Stop bit ,Parity bit) เพื่อให้การรับ และการส่งข้อมูลสามารถจะทำงานให้มีความถูกต้องของข้อมูลมากยิ่งขึ้น

2.1.2 การสื่อสารแบบอนุกรม

Interface หมายถึงการต่อเชื่อมระหว่างอุปกรณ์สองชนิดที่อยู่ในสถานะแวดล้อมในการทำงานที่แตกต่างกัน สำหรับในด้านการสื่อสารข้อมูลหมายถึงการเชื่อมต่อระหว่าง computer หรือ data terminal equipment ชนิดอื่นๆกับอุปกรณ์ภายนอก ในบางครั้งเราเรียกมันว่า I/O port วัตถุประสงค์ประการแรกของการใช้งาน Interface ก็คือการใช้เป็นตัวกลางสำหรับถ่ายโอนข้อมูลกับโลกภายนอก ซึ่งจะต้องมีความสามารถในการ self-protection เพื่อป้องกันการผิดพลาดในระหว่างการถ่ายโอนข้อมูลกับโลกภายนอกอีกด้วย ในการสื่อสารข้อมูลแบบอนุกรมนั้น สิ่งที่เป็นอันตรายจะคาดเดาได้คือ อุปกรณ์ในการแปลงระหว่างการสื่อสารหรือการถ่ายโอนข้อมูลแบบขนานกับการสื่อสารข้อมูลแบบอนุกรม

การรับส่งข้อมูลแบบอนุกรม

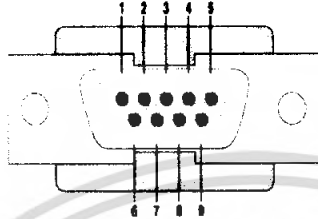
การรับส่งข้อมูลแบบอนุกรม ซึ่งมีข้อดีคือ ใช้สายสัญญาณน้อยและส่งได้เป็นระยะทางไกล สำหรับโรงงานนี้ใช้การติดต่อกับไมโครคอนโทรลเลอร์ รับส่งข้อมูลแบบอนุกรมกับคอมพิวเตอร์ เพื่อทำการเปลี่ยนระดับสัญญาณไฟฟ้าได้มาตรฐานในการรับส่งข้อมูลผ่านทางพอร์ตอนุกรม (Serial Port) ของเครื่องไมโครคอมพิวเตอร์

2.1.3 ความรู้เบื้องต้นเกี่ยวกับมาตรฐาน RS-232

โดยปกติเรื่องไมโครคอมพิวเตอร์จะมีพอร์ตที่เป็นอนุกรมชื่อว่า RS-232 อยู่ในตัวมันเองอยู่แล้ว ซึ่งพอร์ต RS-232 นี้ทำหน้าที่รับและส่งข้อมูลในแบบอนุกรมเรียกว่า Universal Asynchronous Adapter เหตุที่มีชื่อเรียกว่า RS-232 ก็เนื่องมาจาก สมาคมผู้ผลิตอุปกรณ์อิเล็กทรอนิกส์ ของอเมริกา หรือ EIA (RS-232 : Recommended Standard Number 232,EIA: Electronic Industry Association)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ได้กำหนดมาตรฐานของอุปกรณ์สื่อสารแบบอนุกรมเอาไว้ภายใต้ชื่อว่า RS-232 มาตรฐานของรับส่งข้อมูลแบบอนุกรมมีหลายมาตรฐาน แต่ที่นิยมกันมากที่สุดสำหรับเครื่อง ไมโครคอมพิวเตอร์ สำหรับการรับส่งข้อมูลแบบอนุกรมของโครงการนี้นั้นเราจะให้ พอร์ตอนุกรมที่มีลักษณะเป็น 9 pin หรือที่เรียกว่า DB-9



รูป 2.5 แสดง DB-9 ที่ใช้ในการติดต่อผ่านพอร์ตอนุกรม

การกำหนดขาสัญญาณของคอนเน็กเตอร์ (Connector) อนุกรม 9 ขา (DB-9) แสดงในตารางที่ 2.4

ตารางที่ 2.4 แสดงหน้าที่ต่างๆ ของขา DB-9

ขา	ฟังก์ชัน
1	Received Line Signal Detect
2	Received Data
3	Transmit Data
4	Data Terminal Ready
5	Signal Ground
6	Data Set Ready
7	Request to Send
8	Clear to Send
9	Ring Indicator

คำอธิบายการทำงานของแต่ละขาที่ใช้งานของ DB-9

Transmit Data (TD) เป็นสัญญาณที่ส่งออกจาก DTE (หรือตัวไมโครคอมพิวเตอร์) ไปยังโมเด็มหรือต่อกับไมโครคอมพิวเตอร์ตัวอื่น เมื่อไม่มีสัญญาณส่งออกสถานะภาพของลอจิกที่ขานี้มีค่าเท่ากับ "1" หรือเทียบเท่า Stop Bit

Received Data (RD) เป็นทางสัญญาณเข้าไปยัง DTE หรือไมโครคอมพิวเตอร์ เมื่อไม่มีสัญญาณรับเข้ามา ขานี้จะมีสถานะภาพลอจิกเป็น "1"

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Request to Send (RTS) ใช้สำหรับส่งสัญญาณไปยังโมเด็มหรือเครื่องพิมพ์เป็นการเรียกร้องที่จะส่งสัญญาณมาทาง TD สัญญาณนี้จะใช้คู่กับ CTS ที่อุปกรณ์รับ หากได้รับ สัญญาณ RTS จะตรวจสอบตัวเองว่าพร้อมจะรับสัญญาณได้หรือยัง หากพร้อมก็จะส่งสัญญาณออกไปที่สาย CTS

Clear to Send (CTS) เมื่อสายสัญญาณมีอยู่ในสถานะออฟ (Negative Voltage หรือ ลอจิก"1") หมายความว่า อุปกรณ์รับกำลังบอกว่า พร้อมจะรับข้อมูลแล้ว

Data Set Ready (DSR) เมื่อสายสัญญาณมีอยู่ในสถานะออน (ลอจิก"0") จะเป็นการบอกไมโครคอมพิวเตอร์ว่า พร้อมที่จะส่งได้แล้ว

Signal Ground (SG) ทำหน้าที่เป็นระดับแรงดันอ้างอิงสำหรับทุกๆสายสัญญาณจะมีแรงดันเป็น "0" เมื่อเทียบกับสายสัญญาณอื่นๆ

Data Terminal Ready (DTR) คอมพิวเตอร์เปิดสัญญาณนี้ให้ออน(ลอจิก"0") เมื่อพร้อมที่จะติดต่อรับส่งข้อมูล

2.1.4 กอ้งและการเชื่อมต่อของกอ้ง

กอ้งที่ใช้ในการรับภาพใน โครงการนี้เป็นกอ้งเว็บแคมว่า ซึ่งมีคุณสมบัติดังนี้

- กอ้งที่ใช้ในการรับภาพมีความละเอียด 320x240 pixel
- เชื่อมต่อกับคอมพิวเตอร์โดย USB-Port
- ตัวกอ้งมีขนาดเล็ก น้ำหนักเบา
- ระบบ Universal Serial Bus (USB)

ระบบ USB นั้นนับว่าเป็นระบบที่ทันสมัย เนื่องจากรองรับอุปกรณ์ได้มากขึ้น และ ง่ายต่อการติดตั้ง มีความสามารถรองรับ Plug & Play ซึ่งมีคุณสมบัติ ดังนี้

- สามารถลดข้อจำกัดในการต่ออุปกรณ์พ่วงได้มากขึ้นถึง 127 ชิ้น
- ขยายอุปกรณ์มาตรฐานด้วยโครงเวอร์มาตรฐานได้
- สามารถจ่ายไฟฟ้าขนาด 5 Volt ให้แก่อุปกรณ์ที่ต่อพ่วงกับ USB
- "Hot Swapping" สนับสนุนการต่อ ,ถอดออก และรีเซต อุปกรณ์ที่ติดต่อกันโดยไม่ต้อง Reset เครื่อง

Computer

- สามารถส่งถ่ายข้อมูลได้สูงสุดถึง 1.5 Mbit/Sec และ 12 Mbit สัญญาณเสียง และสัญญาณภาพ
- ลดจำนวนสายเคเบิล ท การเชื่อมต่อนั้นก็ง่ายเนื่องจากสายสัญญาณมีแค่ 4 สายสัญญาณ คือ V+ ,D+, D- และ V- โดยสายสัญญาณข้อมูล (D+ และ D-)นั้นจะเป็นแบบ Twist pair
- สายเคเบิลนั้นสามารถนั้นสามารถยาวได้ถึง 5 เมตร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- มีระบบ Suspend เพื่อช่วยในการประหยัดพลังงาน
 - มีการกำหนดค่าตำแหน่งแอดเดรสของ อุปกรณ์ต่างๆ โดยอัตโนมัติ
- ในการเชื่อมต่อของ USB นั้นจะสายเคเบิลแบบ 4 คออร์ ซึ่งมีตำแหน่งขาค้างนี้

- ขา 1 เป็น Voltage +
- ขา 2 เป็น DATA -
- ขา 3 เป็น DATA+
- ขา 4 เป็น Voltage -

2.1.5 สเต็ปมอเตอร์

สเต็ปมอเตอร์ เป็นมอเตอร์ที่ขับเคลื่อนด้วยพัลส์ ลักษณะการขับเคลื่อน จะหมุนรอบแกนได้ 360 องศา มีลักษณะไม่ต่อเนื่อง แต่มีลักษณะเป็นสเต็ป โดยแต่ละสเต็ปจะขับเคลื่อนได้ 1,1.5,1.8 หรือ 2 องศา แล้วแต่โครงสร้างของมอเตอร์ ลักษณะที่ นำมอเตอร์ไปใช้ จะเป็นงานที่ต้องการตำแหน่งแม่นยำ เช่น ระบบขับเคลื่อนหัวแม่พิมพ์ในเครื่องพิมพ์ (PRINTER)ระบบขับเคลื่อนหัวอ่านในเครื่องอ่านบันทึกเทป ระบบขับเคลื่อนตำแหน่งของปากกาใน X-Y PLOTTER เป็นต้น

ข้อดีของสเต็ปมอเตอร์เมื่อเปรียบเทียบกับมอเตอร์กระแสตรง (DC MOTOR)

1. การควบคุมไม่ต้องอาศัยตัวตรวจจับการหมุน
 2. ไม่ต้องใช้แปรงถ่าน ดังนั้นจึงทำให้ไม่มีส่วนที่จะต้องสึกหรอ และปัญหาของการสปาร์ก (ที่เกิดจากหน้าสัมผัสของแปรงถ่านแหวนตัวนำใน โรเตอร์) ที่ทำให้เกิดสัญญาณรบกวน
 3. การควบคุมโดยทางวงจรดิจิทัลหรือไมโคร โพรเซสเซอร์ ทำได้ง่าย และสะดวก
- ชนิดและ โครงสร้างของสเต็ปมอเตอร์

สเต็ปมอเตอร์ที่พบในปัจจุบันมี 3 ลักษณะดังนี้

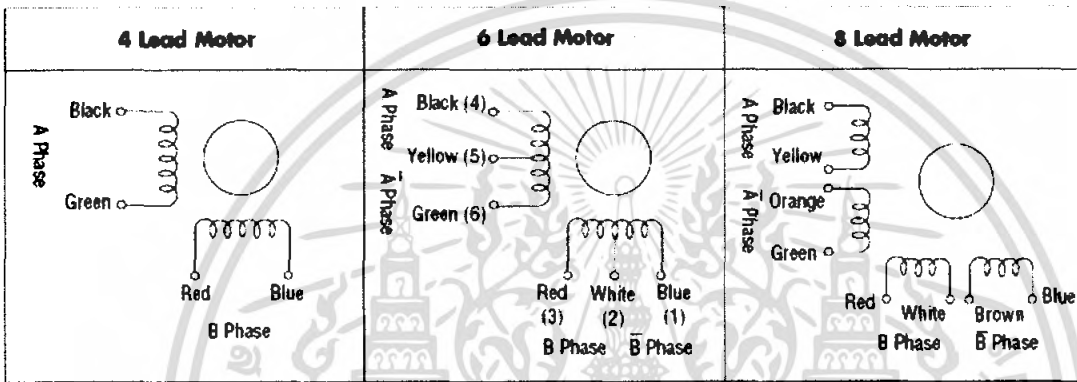
1. แบบแม่เหล็กถาวร(PERMANENT MAGNET_PM) สเต็ปมอเตอร์แบบ PM จะมีสเตเตอร์ (STATOR) ที่พันขดลวดไว้หลายๆ โพล โดยมีโรเตอร์ (ROTOR) เป็นรูปทรง กระบอกฟันเลื่อย และโรเตอร์ทำด้วยแม่เหล็กถาวร เพื่อป้อนไฟกระแสตรง ให้กับขดสเตเตอร์ จะทำให้เกิดแรงแม่เหล็กไฟฟ้าผลักต่อโรเตอร์ ทำให้มอเตอร์หมุนมอเตอร์แบบ PM จะเกิดแรงดูดยึดให้โรเตอร์หยุดอยู่กับที่ แม้จะไม่ได้ป้อนไฟเข้าขดลวด

2. แบบแปรค่ารีลักแตนซ์ (VARIABLE RELUCTANCE- VR) สเต็ปมอเตอร์แบบVR จะมีการหมุนโรเตอร์ได้อย่างอิสระ แม้จะไม่ได้จ่ายไฟให้โรเตอร์ทำจากสารเฟอร์โรแมกเนติก กำลังอ่อนมีลักษณะเป็นฟันเลื่อย รูปทรงกระบอกโดยจะมีความสัมพันธ์ โดยตรงกับจำนวนโพลในสเตเตอร์ แรงบิดที่เกิดขึ้นจะไปหมุนโรเตอร์ ไปในเส้นทางของอำนาจแม่เหล็กที่มีค่ารีลักแตนซ์ต่ำที่สุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตำแหน่งที่จะเกิดแน่นอนและมีเสถียรภาพแต่จะเกิดขึ้นได้หลายๆ จุดคั้งนั้นเมื่อป้อนไฟเข้าขดลวดต่างๆ ในมอเตอร์แตกต่างกันไป ก็ทำให้มอเตอร์ หมุนไปตำแหน่งต่างๆ กัน โรเตอร์ของ VR จะมีความเฉื่อยของโรเตอร์น้อยจึงมีความเร็วรอบสูงกว่ามอเตอร์แบบ PM

3. แบบผสม(HYBRID-H) สเต็ปมอเตอร์แบบ H จะเป็นลูกผสมของ VR กับ PM โดยจะมีสเตเตอร์คล้ายกับที่ใช้ใน VR โรเตอร์มีหมวกหุ้ม ปลายซึ่งมีลักษณะของสารแม่เหล็กที่มีกำลังสูง โดยการควบคุมขนาดรูปร่างของหมวกแม่เหล็กอย่างคิทำให้ได้มุม การหมุนและครั้งน้อยและแม่นยำ ข้อคิก็คือ ให้แรงบิดสูงและมีขนาดกระทัดรัด และให้แรงดูดขั้วโรเตอร์นิ่งกับที่ตอน ไม่จ่ายไฟ



รูป 2.6 แสดง (ก) โครงสร้าง (ข) วงจรเทียบเท่า (equivalent circuit) ของมอเตอร์ ชนิด 4 ขด

ตารางที่ 2.5 แสดงมุมของโรเตอร์เทียบกับกระแสไฟฟ้าที่จ่ายแก่เฟสต่าง ๆ ตำแหน่ง

เฟสที่จ่ายกระแสไฟฟ้า	ϕ_1	$\phi_1\phi_2$	ϕ_2	$\phi_2\phi_3$	ϕ_3	$\phi_3\phi_4$	ϕ_4	$\phi_4\phi_1$
ตำแหน่งโรเตอร์	↑	↗	⇒	↘	↓	↙	←	↖

การสั่งงานควบคุมการหมุนของสเต็ปมอเตอร์

การควบคุมและสั่งงานให้สเต็ปมอเตอร์ทำงาน ไปทีละสเต็ปสามารถทำได้โดยการจ่ายกำลังไฟไปยังขดลวด ในแต่ละขดบนสเตเตอร์ โดยการป้อนจะทำในลักษณะเป็นลำดับหรือเรียกว่า ซีควเอนเชียลในรูปที่ถูกต้อง ซึ่งจะแบบ ได้เป็น 3 รูปแบบ คือ แบบฟูลสเต็ป 1 เฟส แบบฟูลสเต็ป 2 เฟส และแบบครึ่งสเต็ป (half step) ทั้ง 3 แบบนี้ก็จะมีข้อคิและข้อเสยต่างกันออกไป

1.แบบจ่ายกระแสไฟให้เฟสเดียววนเวียนกันไป เรียกว่าแบบฟูลสเต็ป 1 เฟส หรือเรียกว่าแบบเวฟ จะเป็นการกระตุ้นแบบที่ง่ายที่สุด ซึ่งจะทำให้การกระตุ้นขดลวดทีละขดในเวลาหนึ่งๆเรียงกัน ไปตัวอย่างเช่น ขดที่ 1, 2, 3, 4, 1, 2, 3, 4 เป็นลำดับอย่างนี้ หรือ ขด 1, 4, 3, 2, 1, 4, 3, 2 เป็นลำดับกันไป ทั้งนี้ขึ้นอยู่กับทิศทางที่เราต้องให้มอเตอร์หมุนไป วงจรที่นำมากระตุ้นนั้นจะมีราคาค่อนข้างจะถูกกว่าและง่ายกว่า คั้งในรูปของวงจรการจ่ายไฟ ที่อยู่ด้านบนนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เราสามารถเขียนขั้นตอนการทำงานเป็นตารางออกมาได้ดังนี้

ตารางที่ 2.6 แสดงการควบคุมการหมุนของมอเตอร์แบบ wave

สเต็ป	เฟส 1	เฟส 2	เฟส 3	เฟส 4
1	ON			
2		ON		
3			ON	
4				ON
5	ON			
6		ON		

2. แบบจ่ายกระแสไฟให้พร้อมกันทีละ 2 เฟสเรียกหรือ FULL STEP 2 เฟส แบบนี้ก็จะคล้ายกับการกระตุ้นในแบบเวฟแต่จะต่างกันตรงที่ แบบ 2 เฟส จะกระตุ้นทีละ 2 ขด ที่อยู่ใกล้กันในเวลาเดียวกัน และจะเรียงลำดับกันไป ดังเช่นแบบเดียวกับแบบเวฟ จะยกตัวอย่างการกระตุ้นขดลวดในลักษณะ ซีควเอนให้ดูดังนี้ 12,23,34,41,12,23,34,41 เรียงลำดับกันไปเรื่อยๆ หรือจะเป็น 14,43,32,21,14,43,32,21 เรียงกันไปเรื่อยๆเช่นกัน ถ้าจะมากล่าวถึงข้อดีข้อเสียของแบบ 2 เฟส แล้วมีดังนี้ ข้อดี การที่เราจะเพิ่มจำนวนขดลวดที่ถูกกระตุ้นจะทำให้แรงบิดได้มากกว่า แบบเวฟ ซึ่งโรเตอร์จะหมุนด้วยแรงดึงแบบเต็มๆแรงจาก ทั้ง 2 ขดลวดที่กระตุ้นพร้อมกัน ข้อเสีย แบบ 2 เฟส จะกระตุ้นขดลวดนั้นต้องใช้กำลังไฟมากขึ้นเป็น 2 เท่าของแบบเวฟ ก็เป็นไปตามธรรมชาติ ได้อย่างก็ต้องเสียอย่าง นั้นล่ะครับ เราสามารถเขียนลำดับการกระตุ้นของขดลวดแบบ 2 เฟส ได้ดังในตารางต่อไปนี้

ตารางที่ 2.7 แสดงการควบคุมการหมุนของมอเตอร์แบบ Full Step

สเต็ป	เฟส 1	เฟส 2	เฟส 3	เฟส 4
1	ON	ON		
2		ON	ON	
3			ON	ON
4	ON			ON
5	ON	ON		
6		ON	ON	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. แบบจ่ายกระแสไฟฟ้าให้ทีละ 1 เฟส ต่อกับ 2 เฟส เรียก ONE-TWO EXCITATION หรือ HALF STEP เหมือนรูปแสดงของมอเตอร์ แต่แบบนี้จำนวน STEP ทวนเข็มจะเป็นตรงกันข้าม แบบนี้แบบรูปแบบผสมผสานของการกระตุ้นระหว่าง แบบเวฟ กับ แบบ 2 เฟส เพื่อให้จำนวนรอบของสเต็ปให้ มากขึ้นเป็น 2 เท่า ซึ่งในระบบนี้จะทำการกระตุ้นขดลวดเรียงกันไปเรื่อยๆเป็นลำดับ ดังจะยกตัวอย่างต่อไปนี้ 1,12,2,23,3,34,4,41,1,12,2,23,3,34,4,41,1 เป็นลำดับอยู่แบบนี้เรื่อยไป ถ้าเราจะกลับทิศทางการหมุนก็จะไล่เป็นดังนี้ 1,41,4,43,3,32,2,21,1,41,4,43,3,32,2,21,1 เป็นลำดับกันไป

ข้อดี การกระตุ้นแบบนี้จะให้แรงบิดที่เพิ่มมากขึ้น เนื่องจากช่วงสเต็ปที่มีระยะสั้นลง อีกประการหนึ่งแต่ละ สเต็ปเกิดแรงดึงจากขดลวด 2 ขดที่ถูกกระตุ้นพร้อมกันเป็นผลให้ค่าตำแหน่งความถูกต้องมากขึ้นไปด้วย ข้อเสีย เช่นเดียวกับแบบ 2 เฟส ที่ต้องจ่ายกำลังไฟเป็น 2 เท่าของแบบเวฟ หรือจะใช้เท่ากับแบบ 2 เฟส ดังนั้นเราสามารถนำลำดับการทำงานของ แบบครึ่งเฟส ในรูปของตารางได้ดังนี้

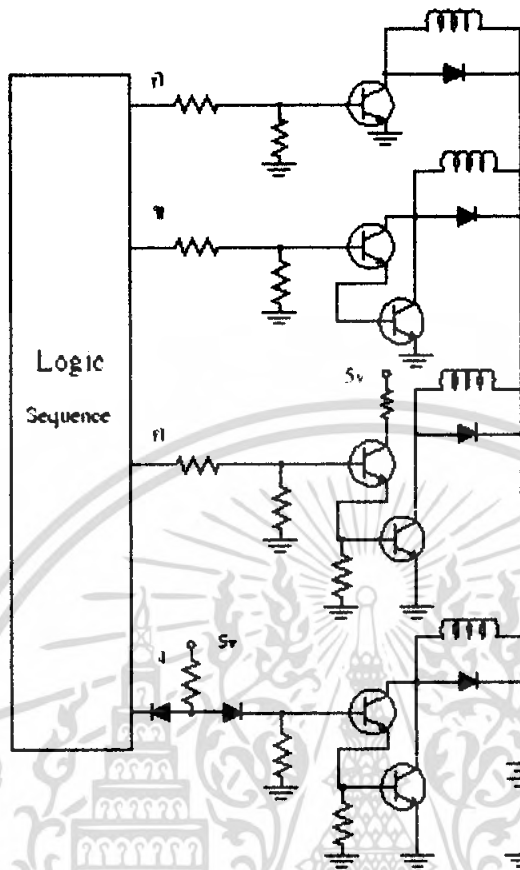
ตารางที่ 2.8 แสดงการควบคุมการหมุนของมอเตอร์แบบ Half Step

สเต็ป	เฟส 1	เฟส 2	เฟส 3	เฟส 4
1	ON			
2	ON	ON		
3		ON		
4		ON	ON	
5			ON	
6			ON	ON
7				ON
8	ON			ON
9	ON			
10	ON	ON		

2.1.6 วงจรขับ (DRIVE)

เมื่อเรารู้ซีแควนซ์ของมันแล้วต่อไปเราก็ต้องมีวงจร DRIVE ให้แก่ STEPPING MOTOR วิธีที่ง่ายที่สุดใน การ ต่อวงจรซีแควนซ์เข้ากับวงจรขับ คือ การต่อโดยตรง ดังเอาท์พุท ก และ ข แต่ถ้ากระแสเอาท์พุท ของวงจรซีแควนซ์ไม่เพียงพอก็ต้องต่อ บัฟเฟอร์ (BUFFER) เพื่อขยายกระแสดึงรูปเอาท์พุท ก และง

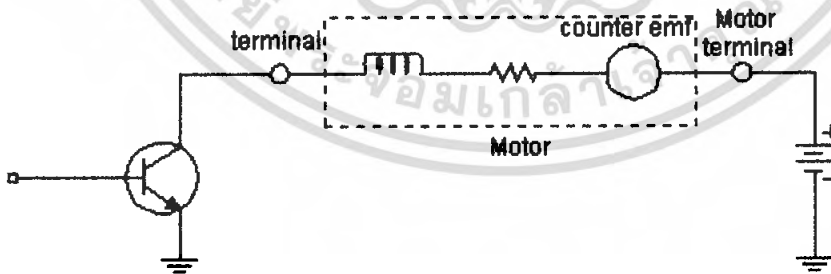
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 2.7 แสดงวิธีขั้วเอาต์พุตของสตีปมอเตอร์

ปัญหาเกี่ยวกับวงจรขับ

ขดลวดของสตีปมอเตอร์เป็น โทลคชนิดตัวเหนี่ยวนำ และมีค่าเปรียบเสมือนผลรวมของความเหนี่ยวนำ (Inductance)อนุกรมกับความต้านทานคังรูป 9



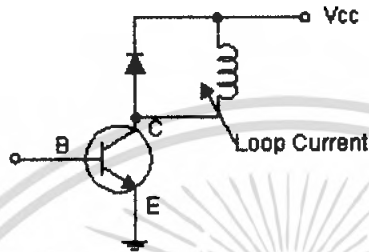
รูป 2.8 วงจรสมมูลย์ (equivalentcircuit) ของสตีปมอเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซัพเพรสเซอร์ (SUPPRESSOR)

เมื่อทรานซิสเตอร์รูป A หยุคนำกระแสจะทำให้เกิดแรงเคลื่อนไฟฟ้าค่าสูงจำนวนหนึ่ง เนื่องจากผลของการเปลี่ยนแปลงของกระแสในตัวเหนี่ยวนำ และแรงเคลื่อนไฟฟ้านี้จะเป็นอันตรายต่อทรานซิสเตอร์ได้มีวิธีป้องกันได้ด้วยวิธีการต่อไปนี้

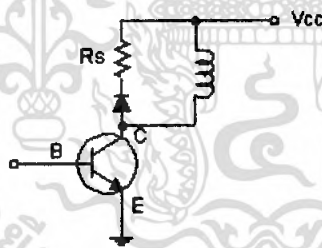
1. ใช้ไดโอดซัพเพรสเซอร์



รูป 2.9 การใช้ไดโอดซัพเพรสเซอร์

กระแสหมุนเวียน CIRCULATING CURRENT จะเริ่มไหลหลังจากทรานซิสเตอร์ หยุคนำกระแสและสวิตคาบคอลเลกเตอร์ จะเท่ากับสวิตคาของแหล่งจ่ายไฟฟ้า ข้อเสียคือ กระแสจะหมุนเวียนอยู่นานและ จะทำให้เกิดแรงบิดห้ามล้อ (BREAKING TORQUE) พลังงานส่วนใหญ่จะสูญเสียเป็นความร้อนในความต้านทานของขดลวด

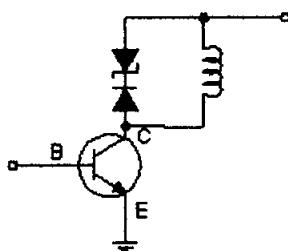
2. ใช้ไดโอดและตัวต้านทานซัพเพรสเซอร์



รูป 2.10 การใช้ไดโอดและตัวต้านทานซัพเพรสเซอร์

ถ้าค่าตัวต้านทาน R_S ยิ่งมากกระแสหมุนเวียนก็จะลดลงเร็วขึ้นแต่สวิตคาของ คอลเลกเตอร์ ในขณะที่กระแสไหลย้อนกลับ จะมีค่าสูงขึ้นพลังงานส่วนใหญ่สูญเสียในตัวต้านทาน R_S

3. ใช้ซีเนอร์ไดโอดซัพเพรสเซอร์

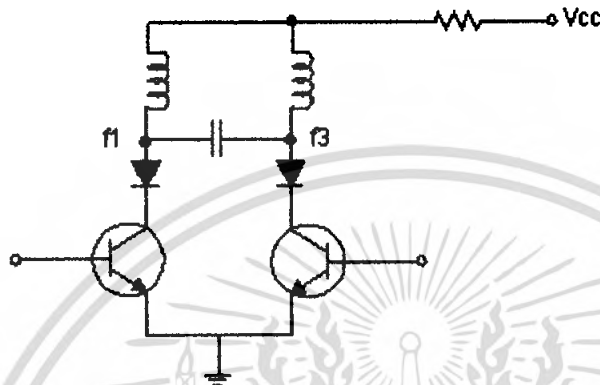


รูป 2.11 การใช้ซีเนอร์ซัพเพรสเซอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอก และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อทรานซิสเตอร์หุคนำกระแส กระแสจะลดลงได้เร็วกว่า 2 แบบแรก และศักดาที่คอลเลกเตอร์ขณะกระแสไหลย้อน กลับจะเท่ากับศักดาของซีเนอร์บวกกับศักดาของแหล่งจ่าย ซึ่งเป็นอิสระต่อกระแสพลังงานส่วนใหญ่สูญเสียในซีเนอร์ไดโอด

4. ใช้ตัวเก็บประจุชัฟเฟอเรเซอร์



รูป 2.12 การใช้ตัวเก็บประจุชัฟเฟอเรเซอร์

จะใส่ตัวเก็บประจุให้ T1 กับ T3 และ T2 กับ T4 เมื่อทรานซิสเตอร์ หุคนำกระแสตัวเก็บประจุ C จะต่อกับทรานซิสเตอร์ โดยผ่านไดโอด และจะดูดกลืนกระแสที่ค่อยๆ ลดลงจากขดลวดของมอเตอร์ เพื่อป้องกันทรานซิสเตอร์เสีย และยังช่วยแคมป์ไฟ ก็ช่วยลดความร้อนที่เกิดขึ้นในขดลวดสเตเตอร์เนื่องจากการแกว่งของโรเตอร์

2.2 ส่วนซอฟต์แวร์

2.2.1 การพัฒนาโปรแกรมด้วยภาษา C

ซีเกิดจากการพัฒนา Microsoft.Net โดยได้รวบรวมจะเด่นและความสามารถของภาษาดั้งเดิมหลายๆ ภาษาเข้าด้วยกัน เช่น VB, C++ และ Java ซึ่งซี เป็นภาษาที่มีโครงสร้างคล้ายภาษา Java โดยเฉพาะความเป็น object ของภาษาซึ่งนอกจากจะรวมเอาจุดเด่นของภาษาต่างๆ มาผสมผสานกันแล้ว ภาษาซียังสร้างจุดเด่นของตัวเองขึ้นมาดังนี้

-เขียนง่าย ซึ่งรูปแบบการเขียนโปรแกรมเป็นแบบ Visual Programming เหมือนกับการเขียนโปรแกรมด้วยภาษา Visual Basic หรือ Delphi

-เป็น Object เนื่องจากภาษาซีมีโครงสร้างคล้ายภาษา Java ดังนั้น โครงสร้างของภาษาจึงเป็นแบบ Object ที่ควมถ้วนสมบูรณ์และมีรูปแบบการเขียนโปรแกรมที่เป็นระบบทำให้พัฒนา Application ได้ง่าย

-ประสิทธิภาพสูง นอกจากความเป็น Object เต็มรูปแบบแล้ว ซียังเป็นภาษาที่พัฒนาขึ้นมาภายใต้ .NET Framework นั่นคือสามารถทำงานภายใต้เทคโนโลยี .NET ได้อย่างสมบูรณ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

-สามารถทำงานระดับลึก ภาษาซีมีความสามารถเกี่ยวกับหน่วยความจำรวมถึงระบบคอมพิวเตอร์ในระดับลึก(ซึ่งแตกต่างกับ Java)

-ทำงานแบบ multi-tread ซึ่งจะช่วยให้โปรแกรมที่เขียนด้วยซีสามารถทำงานได้หลายงานในเวลาเดียวกัน

2.2.2 C++ และ C++บิลเดอร์

C++(ซี-พลัส-พลัส) และ C++บิลเดอร์ (C++Builder : ซี-พลัส-พลัส บิลเดอร์) ต่างก็เป็นภาษาคอมพิวเตอร์ซึ่งใช้สำหรับเขียน โปรแกรมเพื่อสั่งให้คอมพิวเตอร์ทำงานต่างๆ ตามที่เราต้องการ C++ และ C++บิลเดอร์ มีค่าและกฎเกณฑ์ในการเขียนโปรแกรมเหมือนกันเกือบทุกประการแต่มีวิธีเขียนโปรแกรมต่างกัน กล่าวคือ C++ ใช้วิธีเขียนโปรแกรมด้วยอักษรและเครื่องหมาย แต่ C++บิลเดอร์ ใช้วิธีเขียนโปรแกรมแบบวิซวล ซึ่งวิธีการของ C++บิลเดอร์ จะทำให้สามารถเขียนโปรแกรมได้ง่ายและเร็วกว่าวิธีการของ C++

C++บิลเดอร์ เป็น ANSI C++ (เอ็นซี ซี-พลัส-พลัส) ซึ่งหมายความว่า C++บิลเดอร์ เป็นภาษา C++ ที่เป็นไปตามมาตรฐาน ANSI (American National Standard Institute – สถาบันมาตรฐานแห่งชาติของสหรัฐอเมริกา ทำหน้าที่กำหนดมาตรฐานต่างๆ รวมทั้งภาษา C++ ด้วย)

C++ Builder ผลิตโดยบริษัท Inprise Corporation ซึ่งเปลี่ยนชื่อมาจาก บริษัท Borland International, Inc. ประเทศสหรัฐอเมริกา C++ Builder มีชื่อเต็มว่า Borland C++ Builder และนิยมเรียกสั้นๆ ว่า C++ Builder

วิธีเขียนโปรแกรมด้วย C++บิลเดอร์ ทำได้โดยการนำคอมโพเนนต์ที่เรามองเห็น มาวางในฟอร์มแล้วกำหนดลักษณะและการทำงานให้กับคอมโพเนนต์นั้น ในบางโปรแกรมอาจจะต้องเขียนคำสั่งควบคุมการทำงานของโปรแกรมเพิ่มเติม ต่อจากนั้นจึงคอมไพล์(compile) และรัน(run) โปรแกรมนั้นได้ ผลจากการคอมไพล์โปรแกรมจะได้ไฟล์ชนิด .EXE ซึ่งสามารถนำไปรันในคอมพิวเตอร์เครื่องอื่นได้อย่างอิสระโดยไม่ต้องเกี่ยวข้องกับ C++บิลเดอร์ อีกเลย

2.2.3 การเขียนโปรแกรมแบบ Visual

Visual หมายถึงภาพที่เรามองเห็น ดังนั้น การเขียนโปรแกรมแบบ Visual จึงหมายถึง การเขียนโปรแกรมด้วยภาพ หรือการเขียน โปรแกรมด้วยสิ่งที่เรามองเห็น ส่วนประกอบเบื้องต้นสำหรับใช้เขียน โปรแกรมแบบ Visual มี 4 รายการดังนี้

1. Form

2. Component

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.คำสั่งลักษณะการทำงาน

4.คำสั่งสำหรับควบคุมการทำงาน

Object

Object แต่ละตัว จะมีลักษณะเฉพาะตัว ที่ไม่เหมือนกัน เช่น Button จะมีลักษณะเป็นปุ่มกด รูปสี่เหลี่ยม แตกต่างจาก Radio Button ซึ่งมีลักษณะเป็นปุ่มวงกลม มีข้อความด้านขวามือ และหน้าที่ก็ยังคงแตกต่างกันเป็นต้น

ใน C++ Builder เราสามารถกำหนดค่า Property โดยผ่านเครื่องมือที่เรียกว่า Object Inspector ซึ่งกำหนดค่าได้ 3 ลักษณะดังนี้

- 1.การกำหนดโดยอิสระ เช่น Name, Caption
- 2.การกำหนดจากตัวเลือกที่มีให้ ใน Object Inspector เช่น Cursor, Color
- 3.การกำหนดค่าจาก ไดอะล็อกบ็อกซ์ เช่น Font

Method

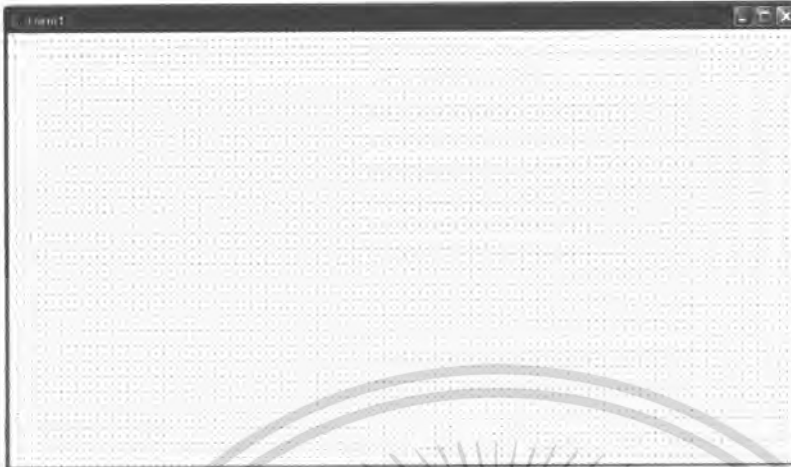
นอกจากการใช้ Property ในการแยกลักษณะเฉพาะ ของ Object แต่ละตัวแล้ว สิ่งหนึ่งที่ Object จะต้องมีนั่นคือ Method นั่นเอง Method คือความสามารถอย่างหนึ่งของ Object ซึ่งถูกเรียกใช้งานตอนที่เราเขียน โปรแกรม ซึ่ง Object แต่ละชนิด จะมี Method ที่แตกต่างกันไป

Event

C++ Builder นั้นเป็นการเขียน โปรแกรมแบบ Event Driven คือใช้เหตุการณ์ต่างๆ ที่เกิดขึ้นมากระทำการต่างๆ ซึ่งสิ่งที่เราจะต้องพบเจอในขณะที่เขียน โปรแกรมก็คือ การจัดการกับเหตุการณ์แต่ละแบบ ซึ่งเราเรียกเหตุการณ์เหล่านั้นว่า Event

Form และ Component

ในการเขียน โปรแกรม C++ Builder เราจะเห็นสิ่งแรกที่เราต้องเจอคือ Form ซึ่งเป็นหน้าต่างหลัก ซึ่งเราใช้ติดต่อกับผู้ใช้งาน การรับข้อมูล หรือการแสดงผล การทำงานให้ผู้ใช้งาน ได้รับรู้และเข้าใจ เราได้นำเอาสิ่งต่างๆ ที่เรียกว่า Component นำมาวางบน Form ซึ่ง component เหล่านั้นก็คือ สิ่งที่จะทำหน้าที่ติดต่อกับผู้ใช้งาน โดยตรง โดยทั้งหมดจะวางอยู่บน Form



รูป 2.13 ฟอรั่มที่ใช้ในการเขียนโปรแกรม

Property ของ Form

เราจะพบว่า มันมี Property อยู่หลายตัว สามารถมองเห็นได้จากภายนอก Property บางตัวนั้น เป็นการกำหนดพฤติกรรมการทำงาน หรือทำงานอยู่เบื้องหลังก็มี โดยมี Property ที่สำคัญดังนี้

- ActiveControl เป็นการกำหนดว่าจะให้ Control ตัวใด Active เมื่อ Form ทำงานครั้งแรก
- Align เป็นรูปแบบการจัดวาง Form ในอยู่ในแนวใด ของจอภาพหรือ Form แม่ ได้แก่ ซิด ด้านบน ด้านล่าง ด้านซ้าย ด้านขวา เต็มจอภาพหรือ Form แม่ และ ไม่จัดแนวระดับ
- AlphaBlend เป็นการกำหนดให้แสดง Form แบบ มัวๆ มองไม่ค่อยชัดเจน หากกำหนดเป็น True ความชัดเจนขึ้นอยู่กับที่กำหนดระดับความชัดเจนที่ AlphaBlendValue
- AlphaBlendValue ความชัดเจนของการแสดง Form แบบ AlphaBlend โดยค่าความชัดเจนอยู่ที่ 0 - 255 โดยถ้าปรับเป็น 0 จะไม่เห็น Form เลย และปรับเป็น 255 เห็น Form ชัดเจนที่สุด
- Anchors เป็นการจัด จุดยึด กับจอภาพหรือ Form แม่ เมื่อจอภาพหรือ Form แม่มีการเปลี่ยนแปลงขนาด โดยจะยึดที่จุดใดเป็นหลัก ได้แก่ ด้านซ้าย ด้านขวา ด้านบน และด้านล่าง
- AutoScroll ถ้ากำหนดค่าเป็น True จะทำให้ Delphi เติม Scrollbar ให้โดยอัตโนมัติทุกครั้ง ที่มีการปรับขนาดของ Form ให้เล็กลงจนไม่สามารถเห็น Component ต่างๆ ได้ครบทั้งหมด
- AutoSize ถ้ากำหนดเป็น True จะทำให้มีการปรับขนาดของ Form ให้พอดี Component ต่างๆ ที่อยู่ภายใน โดยอัตโนมัติ
- BiDiMode เป็นการกำหนด โหมด bi-directional ให้กับ Form
- BorderIcons เป็นการกำหนด Icon บนแถบ Title barว่าจะให้แสดงหรือไม่ ได้แก่ ปุ่มย่อ ขยาย ขนาดของ Form ปุ่มปิด Form และปุ่ม Help
- BorderStyle เป็นลักษณะของขอบของ Form ซึ่งสามารถกำหนดได้ว่า จะยอมให้มีการปรับ

ขนาดของ Form หรือเป็น Form ที่มีขนาดคงที่ หรือจะเป็น หน้าต่างแบบ ใดอะก็อก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Caption เป็นข้อความที่แสดงบน Title bar ของ Form
 - ClientHeight เป็นขนาดความสูงของ Client Area ในหน่วย พิกเซล
 - ClientWidth เป็นขนาดความกว้างของ Client Area ในหน่วย พิกเซล
 - Color เป็นการกำหนดสีพื้นของ Form
 - Constraints เป็นการกำหนดเพื่อจำกัดขนาด กว้างสูงสุด ความกว้างต่ำสุด ความสูงสูงสุด และความสูงต่ำสุดให้กับ Form
 - Ctl3D เป็นการกำหนดการแสดงผลของ Form ว่าจะแสดง แบบ 3D หรือไม่
 - Cursor เป็นลักษณะของ Pointer ของ Mouse เมื่อเราเลื่อนวางเหนือ Form ปกติเป็นรูปลูกศร แต่เราสามารถเปลี่ยนเป็นรูปอื่นๆ ได้เช่น รูปมือ รูปนาฬิกาทราย เป็นต้น
 - DefaultMonitor กำหนดรูปแบบการควบคุมให้กับ Form โดยมีดังนี้ dmDesktop จะไม่มีการกำหนดตำแหน่งการแสดงผล dmPrimary Form จะถูกกำหนดตำแหน่งการแสดงผลตาม List ใน Global screen object's Monitors property ส่วน dmMainForm จะกำหนดเหมือนกับ Application Main Form ทั่วไป และ dmActiveForm จะถูกกำหนดให้เหมือนกับ Form ที่กำลังแสดงอยู่
 - Enabled เป็นการควบคุมการ Active ของ Form ว่าจะให้ Active หรือไม่
 - Font เป็นการกำหนดรูปแบบ Font ตัวอักษร ให้กับ ทุก Component ที่อยู่ใน Form
 - FormStyle เป็นการกำหนดลักษณะของ Form เช่น เป็น Form เดี่ยว (SDI) หรือ Form แม่ ที่มี Form ลูก อยู่ภายใน ได้อีก (MDI)
 - Height เป็นขนาดความสูงของ Form ในหน่วย พิกเซล
 - HelpContext เป็นการกำหนดการเชื่อมโยงไปยัง หมายเลข ของ Help Context หากไม่มีการเชื่อมโยง กำหนดเป็น 0
 - HelpFile เป็นการกำหนด File Help ให้กับ Form
 - HelpKeyword เป็นการกำหนด Keyword สำหรับ Help
 - HelpType เป็นการกำหนดชนิดของ Help ว่าจะเป็น Keyword หรือ Context
 - Hint เป็นการกำหนดคำแนะนำ กำกับ Label เมื่อ Mouse เคลื่อนผ่าน
 - HorzScrollBar เป็นการกำหนดคุณสมบัติของ ScrollBar ในแนวนอน
 - Icon เป็นการกำหนดรูปภาพ ที่จะแสดงเป็น Icon เมื่อ Form ขุบลง และแสดงบน Title bar
- ด้วย
- Left เป็นพิกัดในแกน X ตำแหน่งมุมบน ซ้ายของหน้าจอ
 - Menu เป็นการกำหนด Main Menu ให้กับ Form

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

-Name เป็นชื่อที่ใช้เรียก Form โดยปกติ Delphi 7 จะตั้งให้เป็น Form1 แต่เรามักจะเปลี่ยนชื่อให้จดจำได้ง่าย และสื่อความหมาย หากไม่เปลี่ยนชื่อ จะทำให้ จดจำได้ยากและสับสน เมื่อ โปรแกรมที่เขียนมีขนาดใหญ่ และใช้ Form หลาย Form

-ParentBiDiMode กำหนดว่าจะให้ใช้ Mode bi-directional จาก Object แม่หรือไม่

-ParentFont เป็นการกำหนดว่าจะให้ใช้ รูปแบบ Font จาก Object แม่หรือไม่

-PixelsPerInch เป็นการกำหนด ความละเอียดของ Form ว่าจะใช้ความละเอียดกี่ Pixel ต่อนิ้ว

-PopupMenu เป็นการกำหนด Popup Menu เมื่อมีการ Click Mouse ปุ่มขวา

-Position เป็นการกำหนดตำแหน่งการแสดงผลของ Form ครั้งแรก เมื่อเรียกขึ้นมาใช้งาน เราสามารถกำหนดตำแหน่ง ตามใจตามที่เรต้องการ ได้

-PrintScale เป็นการกำหนด Scale การพิมพ์ ออกทางเครื่องพิมพ์

-ShowHint กำหนดว่าจะให้แสดงข้อความแนะนำหรือไม่

-Tag เป็นการกำหนดลำดับหมายเลขของ Component

-Top เป็นพิกัดในแกน Y ตำแหน่งมุมบน ซ้ายของหน้าจอ

-TransparentColor ถ้ากำหนดเป็น True จะทำให้พื้นหลังเป็นแบบ โปร่งใส ถ้ากำหนดเป็น False จะทำให้พื้นหลังเป็นแบบทึบ

-TransparentColorValue เป็นการกำหนดสี สำหรับการแสดงแบบ TransparentColor

-VetScrollBar เป็นการกำหนดคุณสมบัติของ ScrollBar ในแนวตั้ง

-Visible เป็นการกำหนดว่าจะให้ แสดง Form หรือไม่ หากกำหนดเป็น True จะแสดง

-Width เป็นขนาดความกว้างของ Form ในหน่วย พิกเซล

-WindowState เป็นลักษณะของ Form เมื่อแสดงผล ว่าจะเป็น Form ปกติ ย่อให้เล็กเป็น Icon หรือ ใหญ่เต็มหน้าจอ

Method ของ Form

Form เป็น Object ดังนั้น มันจึงมี Method ที่สำคัญในการทำงานดังนี้

-Show เป็น Method ที่ใช้ในการแสดง Form ขึ้นมา

-ShowModal เป็น Method ที่ใช้ในการแสดง Form ขึ้นมา แบบ Modal หมายความว่า Modal

-Form คือ Form ที่ไม่ยอมให้ Form แสดงขึ้นมาเหนือตัวเอง Form อื่นๆ จะไม่ Active

ในขณะที่ Modal Form กำลังแสดงอยู่ ดังนั้น จึงต้องปิด Form แบบ Modal เสียก่อน Form อื่นๆ จึงจะ Active และแสดง Form อื่นๆ ได้

-Close เป็น Method ที่ใช้ในการ ปิด Form

-Hide เป็น Method ที่ใช้ในการ ซ่อน Form

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Event ของ Form

Form มี Event ต่างๆ ที่ใช้ในการตรวจจับ เหตุการณ์ ต่างๆที่เกิดขึ้น และเราใช้บ่อยๆ มีดังนี้

- OnActive เป็นเหตุการณ์ที่เกิดขึ้นเมื่อ มีการย้ายการทำงานกลับเข้ามายัง Form
- OnCanResize เป็นเหตุการณ์ที่เกิดขึ้นเมื่อ ขณะที่กำลังเปลี่ยนขนาด Form
- OnClick เป็นเหตุการณ์ที่เกิดขึ้นเมื่อ มีการ Click Mouse ยัง ไม่ยกนิ้วขึ้น
- OnClose เป็นเหตุการณ์ที่เกิดขึ้นเมื่อ มีการปิด Form
- OnCloseQuery เป็นเหตุการณ์ที่เกิดขึ้นเมื่อ ก่อนจะเริ่มเปิด Form อาจจะมีการยืนยันก่อนการ

เปิด ---Form หรือกระทำการอื่นๆ ขึ้นอยู่กับการทำงานของผูเขียน

- OnConstrainedResize เป็นเหตุการณ์ที่เกิดขึ้นเมื่อ มีการเปลี่ยนแปลงค่าควบคุมขนาดของ

Form

- OnContextPopup เป็นเหตุการณ์ที่เกิดขึ้นเมื่อ มีการแสดง Context ขึ้นมา
- OnCreate เป็นเหตุการณ์ที่เกิดขึ้นเมื่อ มีการขออนุญาตความจำจาก Windows เพื่อเริ่มสร้าง

Form

- OnDbClick เป็นเหตุการณ์ที่เกิดขึ้นเมื่อ มีการ Double Click Mouse ที่ Form
- OnDeactive เป็นเหตุการณ์ที่เกิดขึ้นเมื่อ มีการย้ายการทำงานออกไปจาก Form
- OnDestroy เป็นเหตุการณ์ที่เกิดขึ้นเมื่อ ปิด Form เสร็จเรียบร้อยแล้ว และได้คืน

หน่วยความจำที่ใช้ จากการสร้าง Form ให้กับ Windows แล้ว

- OnDockDrop เป็นเหตุการณ์ที่เกิดขึ้นเมื่อ เป็นเหตุการณ์ที่เกิดขึ้นเมื่อมีการ DockDrop
- OnDragDrop เป็นเหตุการณ์ที่เกิดขึ้นเมื่อ DragDrop Mouse
- OnDockOver เป็นเหตุการณ์ที่เกิดขึ้นเมื่อ เป็นเหตุการณ์ที่เกิดขึ้นเมื่อกำลัง Dock
- OnEndDock เป็นเหตุการณ์ที่เกิดขึ้นเมื่อ เป็นเหตุการณ์ที่เกิดขึ้นจบการ Dock
- OnGetSiteInfo เป็นเหตุการณ์ที่เกิดขึ้นเมื่อ มีการส่งค่าข้อมูลการ Dock กลับมา
- OnHelp เป็นเหตุการณ์ที่เกิดขึ้นเมื่อ มีการเริ่มต้น ใช้ Help
- OnHide เป็นเหตุการณ์ที่เกิดขึ้นเมื่อ มีการซ่อน Form
- OnKeyDown เป็นเหตุการณ์ที่เกิดขึ้นเมื่อ มีการกดปุ่ม Key Board ขาลง
- OnKeyPress เป็นเหตุการณ์ที่เกิดขึ้นเมื่อ มีการกดปุ่ม Key Board ขณะที่ยังไม่ยกขึ้น
- OnKeyUp เป็นเหตุการณ์ที่เกิดขึ้นเมื่อ มีการกดปุ่ม Key Board ขาขึ้น
- OnMouseDown เป็นเหตุการณ์ที่เกิดขึ้นเมื่อ มีการกดปุ่ม Mouse ขาลง
- OnMouseMove เป็นเหตุการณ์ที่เกิดขึ้นเมื่อ มีการเคลื่อน Mouse ผ่าน Form
- OnMouseUp เป็นเหตุการณ์ที่เกิดขึ้นเมื่อ มีการกดปุ่ม Mouse ขาขึ้น
- OnMouseWheel เป็นเหตุการณ์ที่เกิดขึ้นเมื่อ มีการเลื่อนล้อ Wheel ของ Mouse

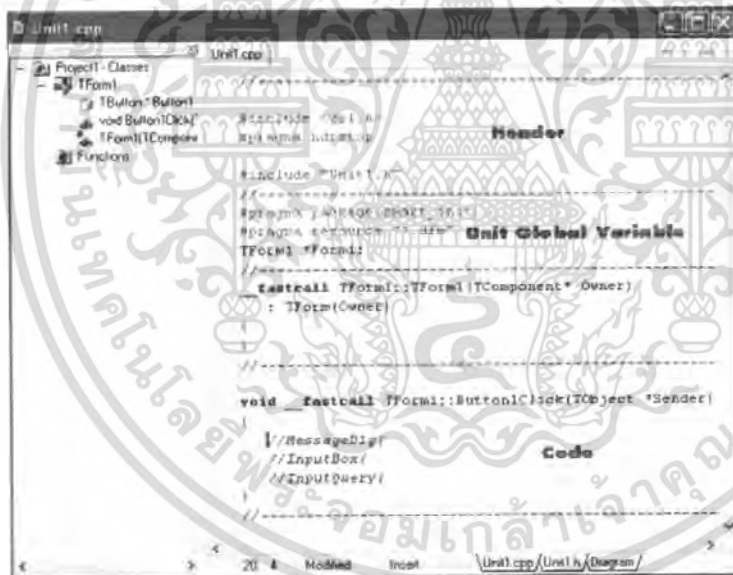
เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อใช้ในการศึกษาเท่านั้น ไม่ควรนำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- OnMouse Wheel Down เป็นเหตุการณ์ที่เกิดขึ้นเมื่อ มีการกด ล้อ Wheel ของ Mouse ขาลง
- OnMouse Wheel IUp เป็นเหตุการณ์ที่เกิดขึ้นเมื่อ มีการกด ล้อ Wheel ของ Mouse ขาขึ้น
- OnPaint เป็นเหตุการณ์ที่เกิดขึ้นเมื่อ มีการวาด Form
- OnResize เป็นเหตุการณ์ที่เกิดขึ้นเมื่อ มีการปรับขนาดของ Form
- OnShortCut เป็นเหตุการณ์ที่เกิดขึ้นเมื่อ มีการ กด Key ลัดต่างๆ
- OnShow เป็นเหตุการณ์ที่เกิดขึ้นเมื่อ มีการแสดง Form
- OnStartDock เป็นเหตุการณ์ที่เกิดขึ้นเมื่อ มีการเริ่ม Dock Form
- OnUnDock เป็นเหตุการณ์ที่เกิดขึ้นเมื่อ มีการยกเลิกการ Dock Form

โครงสร้าง Unit

เมื่อเราสังเกตที่ File Unit ที่ C++ Builder สร้างให้เราเราสามารถแบ่ง Code ออกเป็นส่วนๆ ดังนี้

1. Unit.cpp



รูป 2.14 Unit.cpp

- Header ใช้ประกาศ File ส่วนหัวที่จะนำมาพร้อมกับ โปรแกรมที่เราจะเขียนขึ้น อันได้แก่ File *.h ต่างๆ
- Unit Local Variable ส่วนนี้ใช้ประกาศ ตัวแปรแบบ Global สำหรับ Unit นั้นๆ ไม่สามารถให้ Unit อื่นๆ ใช้ร่วมได้ ใช้ได้เพียง Unit นี้เท่านั้น
- Code ใช้สำหรับเขียนคำสั่งต่างๆ และ Function หรือ Method ต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. Unit.h

```

#ifndef UNIT_H
#define UNIT_H

//
// Header Include
#include <QtWidgets.h>
#include <Unit.h>
#include <QtCore.h>
#include <QtGui.h>

//
// Class Define
class TForm : public TForm
{
    Q_OBJECT
public:
    TForm(QWidget *parent = 0);
    void fastcall ButtonClick(TObject *Sender);
private:
    // User declarations
public:
    // User declarations
    fastcall TForm(TComponent* Owner);
};

//
// Extern
extern PACKAGE TForm *Form1;

#endif

```

รูป 2.15 Unit.h

-Unit Define เป็นส่วนเริ่มประกาศ Unit สร้าง Class Unit นั้นๆ

-Header Include ใช้ประกาศ File ส่วนหัวที่จะนำมาพร้อมกับ โปรแกรมที่เราจะเขียนขึ้น อันได้แก่ File *.h, *.hpp ต่างๆ ซึ่งใช้ประกอบการสร้าง Form และ Object ต่างๆ

-Class Define เป็นส่วนที่ใช้สร้าง Class สำหรับ Unit หรือ Form นั้นๆ อันประกอบไปด้วย 4 ส่วนดังนี้

__published ส่วนนี้ใช้ประกาศ Object, Method, Event และตัวแปร ต่างๆ ที่ใช้ใน Unit นั้นๆ Support VCL เท่านั้น

private ส่วนนี้ใช้ประกาศ Object, Method, Event และตัวแปร ต่างๆ ที่ใช้ใน Unit นั้นๆ โดย Unit อื่นๆ ไม่สามารถมองเห็นได้

public ส่วนนี้ใช้ประกาศ Object, Method, Event และตัวแปร ต่างๆ ที่ Unit หรือ Module อื่นๆ สามารถมองเห็นได้

protected ส่วนนี้ใช้ประกาศ Object, Method, Event และตัวแปร ต่างๆ ที่ Unit หรือ Module อื่นๆ ที่สืบทอดมาจาก Class นี้เท่านั้นที่สามารถมองเห็นได้

Extern เป็นส่วนที่ประกาศเพื่อให้ Unit นั้นสามารถใช้งานได้ และมองเห็นได้ โดย Class และ Unit อื่นๆ พูกว่าๆ เป็นการส่ง Class ออกไปใช้งานนั่นเอง

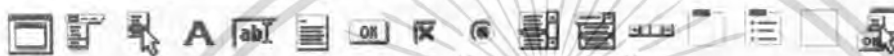
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

C++ Builder Component

การพัฒนาโปรแกรม หรือ Application Software ที่ผ่านมาก็คือ การที่เรามีเครื่องมือช่วยให้ การเขียนโปรแกรม และสร้าง Application อย่าง C++ Builder ที่ช่วยให้เราสามารถออกแบบ เขียน โปรแกรม ทดสอบ และแก้ไขสิ่งที่ได้สร้างขึ้นมา นั้น ได้อย่างรวดเร็ว ยืดหยุ่น ซึ่งเมื่อผ่านไปหลาย Version เข้า ก็มี Component ชนิดต่างๆ มากมายแทบใช้งานไม่หมด

C++ Builder ได้จัดแบ่ง Component ต่างๆ ออกเป็นกลุ่มให้เลือก ใช้งาน ได้ตามลักษณะงาน ซึ่งมี มากมายถึง 29 กลุ่ม โดยแยกกันไว้ใน Component Palette ดังนี้

1. Standard เป็นกลุ่ม Component มาตรฐานซึ่งมีการใช้งานทั่วไปใน Windows



รูป 2.16 component กลุ่ม standard

2. Additional เป็นกลุ่ม Component เพิ่มเติมพิเศษ ช่วยเสริมการทำงาน ของ Component มาตรฐาน ช่วยให้ Application ใช้งานสะดวก และสวยงามน่าใช้



รูป 2.17 component กลุ่ม Additional

3. Win32 เป็นกลุ่ม Component ซึ่งเกิดมาพร้อมกับระบบปฏิบัติการ 32 บิต ของ Windows 95, 98, NT, 2000, Me, XP และ 2003 Server



รูป 2.18 component win32

4. System เป็นกลุ่ม Component ที่สร้างเพื่อติดต่อกับ การทำงานภายใน ของระบบคอมพิวเตอร์ที่เรา ใช้งาน



รูป 2.19 component กลุ่ม system

5. Data Access เป็นกลุ่ม Component ที่ใช้ในการติดต่อฐานข้อมูล



รูป 2.20 component กลุ่ม data access

6. Data Controls เป็นกลุ่ม Component ที่ใช้สร้างส่วนติดต่อผู้ใช้ ใน Application ด้านฐานข้อมูล



รูป 2.21 component กลุ่ม data control

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7.dbExpress เป็นกลุ่ม Component ที่ใช้สร้าง Application ด้านฐานข้อมูลด้วย dbExpress



รูป 2.22 component กลุ่ม dbExpress

8.DataSnap เป็นกลุ่ม Component ที่ใช้สร้าง Application ฐานข้อมูลแบบ multi-tiered



รูป 2.23 component กลุ่ม datasnap

9.BDE เป็นกลุ่ม Component ที่ใช้สร้าง Application เชื่อมต่อกับฐานข้อมูลที่ใช้ Borland Database Engine



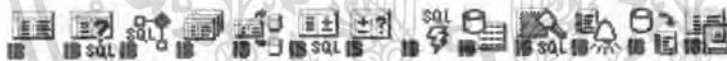
รูป 2.24 component กลุ่ม BDE

10.ADO เป็นกลุ่ม Component ที่สร้างเพื่อรองรับ เทคโนโลยีเชื่อมต่อฐานข้อมูลแบบ ADO ของ Microsoft



รูป 2.25 component กลุ่ม ADO

11.InterBase เป็นกลุ่ม Component ที่สร้างขึ้นมาเพื่อรองรับ เทคโนโลยี การเชื่อมต่อกับฐานข้อมูลแบบ Interbase ของ Borland



รูป 2.26 component กลุ่ม InterBase

12.WebServices เป็นกลุ่ม Component ที่ใช้เขียน Application ฟังก์ชัน Client ที่จะเรียกใช้ WebServices ผ่าน SOAP



รูป 2.27 component กลุ่ม Webservice

13.InternetExpress เป็นกลุ่ม Component ที่ใช้สร้าง Application ที่ใช้งานผ่าน Internet ในรูปแบบใหม่ซึ่งรองรับ XML



รูป 2.28 component กลุ่ม InternetExpress

14.Internet เป็นกลุ่ม Component ที่ใช้สร้าง Application Internet ในฝั่ง Server



รูป 2.29 component กลุ่ม Internet

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

15. WebSnap เป็นกลุ่ม Component ที่ใช้สร้าง Application Internet ในฝั่ง Server



รูป 2.30 component กลุ่ม WebSnap

16. FastNet เป็นกลุ่ม Component ที่ใช้สร้าง Application Internet ในฝั่ง Client



รูป 2.31 component กลุ่ม FastNet

17. Decision Cube เป็นกลุ่ม Component ที่ใช้สร้าง Application ที่ต้องการการวิเคราะห์ข้อมูล ขั้นสูง โดยปกติจะเป็นข้อมูลจากฐานข้อมูลขนาดใหญ่



รูป 2.32 component กลุ่ม Decision Cube

18. QReport เป็นกลุ่ม Component ที่ใช้สร้าง Application ในการรายงานข้อมูล



รูป 2.33 component กลุ่ม QReport

19. Dialogs เป็นกลุ่ม Component ที่นำมาเรียกใช้ Dialogs มาตรฐานของ Windows



รูป 2.34 component กลุ่ม Dialog

20. Win3.1 เป็นกลุ่ม Component ที่นำมาสร้าง Application ที่ต้องการให้ใช้งานร่วมกับ Component ใน Style ของ Windows 3.1 ได้



รูป 2.35 component กลุ่ม Win3.1

21. Samples เป็นกลุ่ม Component ที่ C++ Builder ได้เตรียมเอาไว้ ให้ศึกษาการสร้าง Component ขึ้นมาใช้เอง ซึ่งจะมี Source Code ของ Component นั้นทั้งหมดด้วย โดยจะอยู่ที่ Source\Samples ภายใน Folder ที่ติดตั้ง C++ Builder



รูป 2.36 component กลุ่ม Samples

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

22.ActiveX เป็นกลุ่ม Component ที่ใช้สร้างตามแนวความคิดของ ActiveX Control ของ Microsoft



รูป 2.37 component กลุ่ม ActiveX

23.COM+ เป็นกลุ่ม Component ที่ทำหน้าที่สร้าง Application ตามแนวความคิดของ COM+ (COM PLUS)



รูป 2.368 component กลุ่ม COM+

24.Interbase Admin เป็นกลุ่ม Component ที่ช่วยเพิ่มความสะดวก ในการพัฒนา Application ด้วย Interbase 6



รูป 2.39 component กลุ่ม Interbase Admin

25.Servers เป็นกลุ่ม Component ที่ใช้ Object COM Server



รูป 2.40 component กลุ่ม Servers

26.Indy Clients เป็นกลุ่ม Component ที่ใช้สร้าง Application ในฝั่ง Client ตามแนวความคิดของ Internet Direct หรือ Indy ซึ่งก็คือ Component ประเภท Opensource ซึ่งจะเรียกใช้งาน โปรโตคอล อินเทอร์เน็ต มาตรฐานต่างๆ



รูป 2.41 component กลุ่ม Indy Client

27.Indy Servers เป็นกลุ่ม Component ที่ใช้สร้าง Application ในฝั่ง Server ตามแนวความคิดของ Internet Direct หรือ Indy ซึ่งก็คือ Component ประเภท Opensource ซึ่งจะเรียกใช้งาน โปรโตคอล อินเทอร์เน็ต มาตรฐานต่างๆ



รูป 2.42 component กลุ่ม Indy Servers

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

28. Indy Misc เป็นกลุ่ม Component ที่ช่วยสร้าง Application ตามแนวความคิดของ Internet Direct หรือ Indy ช่วยให้มัลติเพล็กซ์ และหน้าใช้งานมากขึ้น



รูป 2.43 component กลุ่ม Indy Misc

29. Office2k เป็นกลุ่ม Component ที่ช่วยสร้าง Application กับ Microsoft Office 2000



รูป 2.44 component กลุ่ม Office2k

C++ Builder Working Mode

ในการพัฒนา Application ด้วย C++ Builder ซึ่งเราสามารถสร้าง Application ไปพร้อมๆ กับการทดสอบ ปรับแต่ง และแก้ไข จนกว่าจะเป็นที่พึงพอใจนั้น ได้แบ่งการทำงานออกเป็น 3 Mode ได้แก่

Design Mode เป็น Mode ทำงานในขณะที่เรากำลังออกแบบ Application หรือเขียน Form กำหนดค่าของ Property ต่างๆ หรือการเขียนคำสั่งใน code Editor

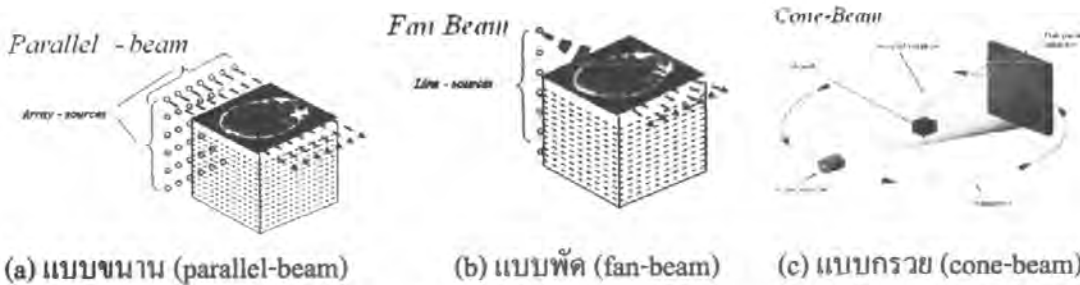
Run Mode เป็น Mode การทำงานเมื่อ Application เริ่มทำงาน โดย Application จะต้องถูก Compile เสียก่อน

Debug Mode เป็น Mode การทำงานที่เรา ใช้เพื่อตรวจสอบการทำงานของ Application นั้น คือจะต้องให้ Application ทำงานก่อน แล้วค่อยตรวจสอบการทำงาน อยู่เบื้องหลัง เพื่อ

2.3 อัลกอริทึมของการสร้างภาพตัดขวางแบบกรวย

อัลกอริทึมที่ใช้ในการสร้างภาพตัดขวางสำหรับทางเดินแสงแบบกรวย วิธีเฟลคแคม (Feldkemp Cone-beam) ซึ่งพิจารณาทางเดินแสงแบบกรวย เป็นกระบวนการสร้างภาพตัดขวางแบบ 2 มิติที่สามารถสร้างภาพตัดขวางของวัตถุได้ครั้งละระนาบ ถ้าวัตถุมีข้อมูลเป็นแบบปริมาตรก็ สามารถทำได้โดยการแบ่งวัตถุออกเป็นหลายๆระนาบแล้วสร้างภาพตัดขวางทีละระนาบ แต่การสร้าง ภาพตัดขวางแบบขนานนี้ยังไม่สามารถนำไปใช้ได้กับข้อมูลจริงได้ ดังนั้นจึงมีการพิจารณาทางเดิน ของแสงใหม่ให้ใกล้เคียงกับความเป็นจริงคือคือทางเดินแสงแบบพัดและทางเดินแสงแบบกรวยดังรูป ที่ 2.45

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



(a) แบบขนาน (parallel-beam)

(b) แบบพัด (fan-beam)

(c) แบบกรวย (cone-beam)

รูปที่ 2.45 เส้นทางการเดินแสง

2.3.1 การสร้างภาพตัดขวางแบบพัด(Fan-beam reconstruction)

กระบวนการสร้างภาพตัดขวางโดยใช้อัลกอริทึมแบบ 3 มิติ(cone-beam reconstruction) พิจารณาทางเดินแสงเป็นแบบกรวย ซึ่งมีลักษณะเหมือนพัดที่ขยขึ้น- สรรอบจุดหมุนที่ตำแหน่งแหล่งกำเนิดแสง (point source) เพื่อให้เกิดความเข้าใจจึงต้องกล่าวถึงวิธีการสร้างภาพแบบพัดเสียก่อนซึ่งถูกแบ่งออกเป็น 2 แบบคือ

1.มุมระหว่างเส้นแสงเท่ากัน(Equiangular ray)

กำหนดให้ $R_\beta(s)$ เป็นข้อมูลโปรเจกชันที่มีทางเดินแสงเป็นแบบพัดดังรูปที่ 2.46 ซึ่ง β เป็นมุมที่ทำกับแกน y พิจารณาเส้นแสง SA ถ้าโปรเจกชันถูกสร้างโดยพิจารณาทางเดินแสงเป็นแบบขนานนั่นคือ SA เป็นเส้นแสงเส้นหนึ่งของโปรเจกชัน $p_\theta(t)$ มีมุมเท่ากับ θ ตำแหน่ง t

$$\theta = \beta + \gamma$$

$$t = D \sin \gamma$$

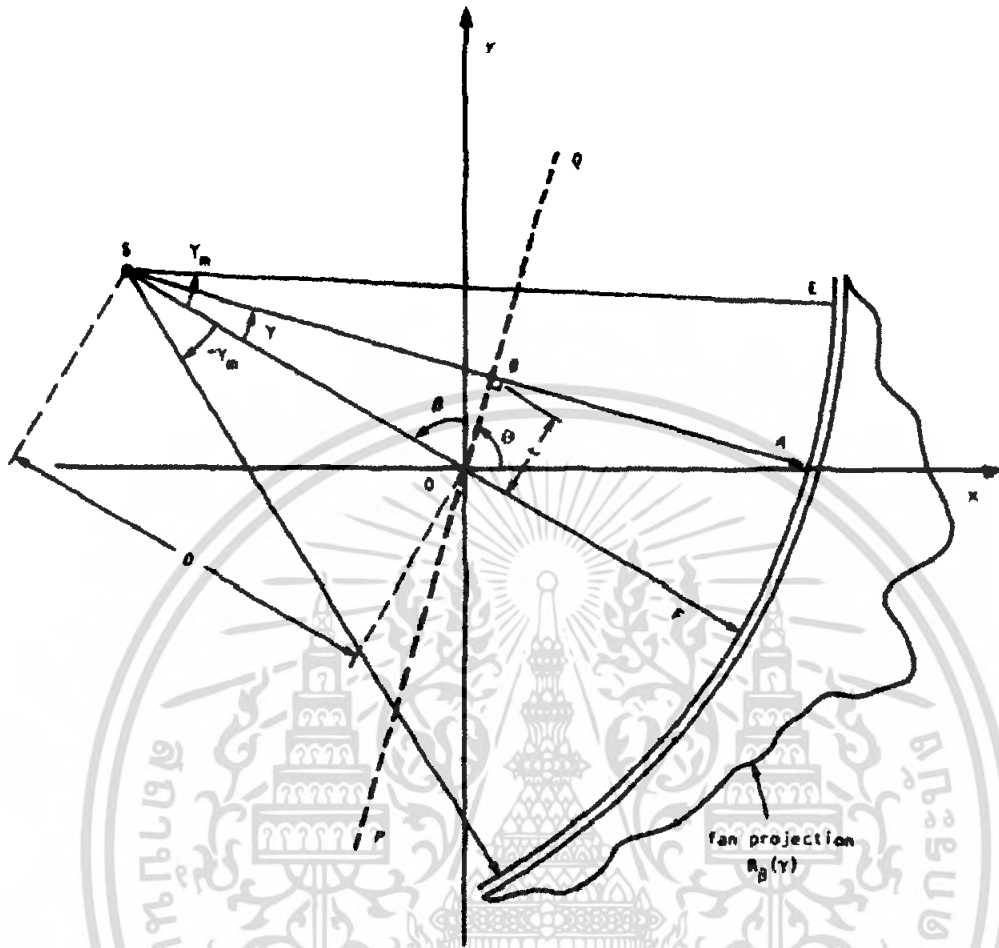
(2.1)

โดย D คือระยะห่างระหว่างแหล่งกำเนิดแสง S ถึงจุดกำเนิด O ความสัมพันธ์ในสมการที่

(2.1) ถูกนำมาใช้ในการพิจารณาให้อยู่ในรูปแบบเดียวกันกับทางเดินแสงแบบขนาน

$$f(x, y) = \int_0^\pi \int_{-r_m}^{r_m} p_\theta(t) \cdot h(s \cos \theta + y \sin \theta) dt d\theta \quad (2.2)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.46 ข้อมูลโปรเจกชันเมื่อพิจารณาทางเค้นแสงแบบพัดที่มีระยะห่างระหว่างหัวรับเท่ากัน
(Equally space collinear detectors)

โดยที่ r_m คือขอบเขตบนแกนคิเทคเตอร์ สมการ (2.2) นี้เป็นสมการการสร้างภาพตัดขวางวิธีฟิลเตอร์เบ็ค โปรเจกชันโดยพิจารณาทางเค้นแสงแบบขนาน ดังนั้นช่วงมุมโปรเจกชันจึงอยู่ระหว่าง 0 ถึง 180 องศา อย่างไรก็ตามการสร้างภาพตัดขวางจากโปรเจกชันที่เป็นแบบพัดนั้นจำเป็นต้องใช้ช่วงมุมที่มากกว่าคือระหว่าง 0 ถึง 360 องศา ดังนั้นเขียนสมการ (2.2) ได้ใหม่คือ

$$f(x, y) = \frac{1}{2} \int_0^{2\pi} \int_{-r_m}^{r_m} p_\theta(t) \cdot h(x \cos \theta + y \sin \theta) dt d\theta \quad (2.3)$$

จากรูปที่ (2.3) ที่พิกัด (x, y) หรือจุด C สามารถเขียนให้อยู่ในระบบพิกัดเชิงขั้ว (r, ϕ) ได้คือ

$$\begin{aligned} x &= r \cos \phi \\ y &= r \sin \phi \end{aligned} \quad (2.4)$$

สมการที่ (2.3) เขียนให้อยู่ในระบบพิกัดเชิงขั้วได้เป็น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

L คือระยะห่างระหว่างแหล่งกำเนิดแสง S ถึงพิกัด (x, y) หรือ (r, ϕ) ในระบบพิกัดทรงกลมเช่นจุด C ในรูป 2.47 ซึ่ง L เป็นฟังก์ชันของ 3 ตัวแปรคือ r, ϕ, β และ γ' เป็นมุมของเส้นแสงที่ผ่านพิกัด (r, ϕ) ซึ่งเขียนความสัมพันธ์ได้คือ

$$L \cos \gamma' = D + r \sin(\beta - \phi)$$

$$L \sin \gamma' = r \cos(\beta - \phi) \quad (2.9)$$

$$L(r, \phi, \beta) = \sqrt{[D + r \sin(\beta - \phi)]^2 + [r \cos(\beta - \phi)]^2} \quad (2.10)$$

และ

$$r' = \tan^{-1} \left[\frac{r \cos(\beta - \phi)}{D + r \sin(\beta - \phi)} \right] \quad (2.11)$$

นำสมการ (2.9) แทนใน (2.8) ได้

$$r \cos(\beta + \gamma - \phi) - D \sin \gamma = L \sin(\gamma' - \gamma) \quad (2.12)$$

และนำไปแทนใน (2.7) จะได้

$$f(r, \phi) = \frac{1}{2} \int_{0-\gamma_n}^{2\pi-\gamma_n} \int R_\beta(\gamma) h(L \sin(\gamma' - \gamma)) D \cos \gamma d\gamma d\beta \quad (2.13)$$

สำหรับฟังก์ชัน h เป็นเทอมของการพิลเตอร์ ซึ่ง $h(t)$ เป็นการแปลงกลับฟูเรียร์ของ $|\omega|$ ในโดเมนของความถี่

$$h(t) = \int_{-\infty}^{\infty} |\omega| e^{j2\pi t \omega} d\omega \quad (2.14)$$

เพราะฉะนั้น

$$h(L \sin \gamma) = \int_{-\infty}^{\infty} |\omega| e^{j2\pi L \sin \gamma \omega} d\omega \quad (2.15)$$

กำหนด ω' ให้เป็น

$$\omega' = \frac{\omega L \sin \gamma}{\gamma} \quad (2.16)$$

สามารถเขียนได้เป็น

$$\begin{aligned} h(L \sin \gamma) &= \left(\frac{\gamma}{L \sin \gamma} \right)^2 \int_{-\infty}^{\infty} |\omega| e^{j2\pi \omega' \gamma} d\omega \\ &= \left(\frac{\gamma}{L \sin \gamma} \right)^2 h(\gamma) \end{aligned} \quad (2.17)$$

สมการที่ (2.13) เขียนได้เป็น

$$f(r, \phi) = \int_0^{2\pi} \frac{1}{L^2} \int_{-\gamma}^{\gamma} R_\beta(\gamma) g(\gamma' - \gamma) D \cos \gamma d\gamma d\beta \quad (2.18)$$

โดยเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$g(\gamma) = \frac{1}{2} \left(\frac{\gamma}{\sin \gamma} \right)^2 h(\gamma) \quad (2.19)$$

ดังนั้นสมการการสร้างภาพตัดขวางวิธีฟิลเตอร์เบ็ค โปรเจกชันทางเดินแสงแบบพัดโดยพิจารณาумระหว่างเส้นแสงเท่ากัน เขียนสมการที่ (2.18) ได้เป็น

$$f(r, \phi) = \int_0^{2\pi} \frac{1}{L^2} Q_\beta(\gamma') d\beta \quad (2.18)$$

โดยที่

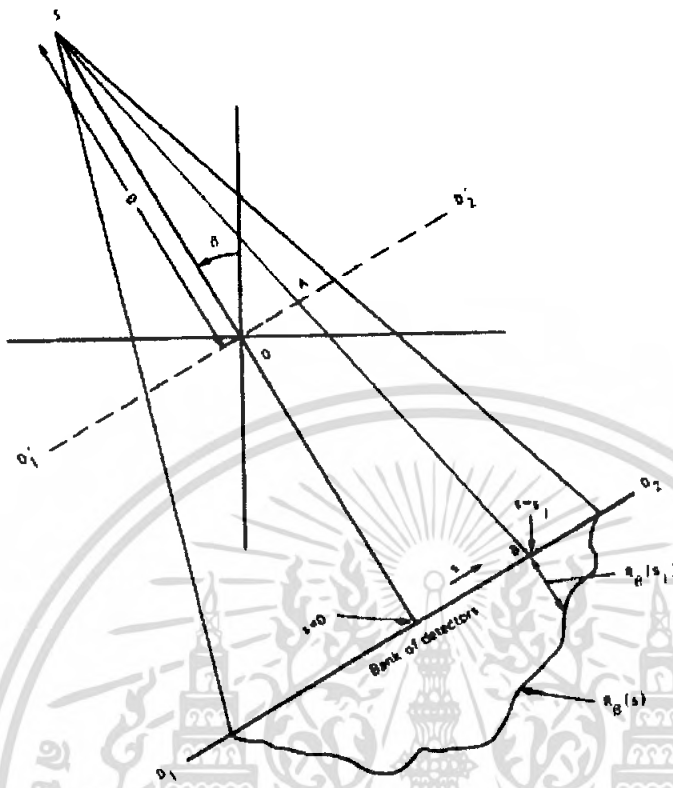
$$Q_\beta(\gamma) = R'_\beta(\gamma') * g(\gamma) \quad (2.19)$$

$$R'_\beta(\gamma') = R_\beta(\gamma) \cdot D \cdot \cos \gamma \quad (2.20)$$

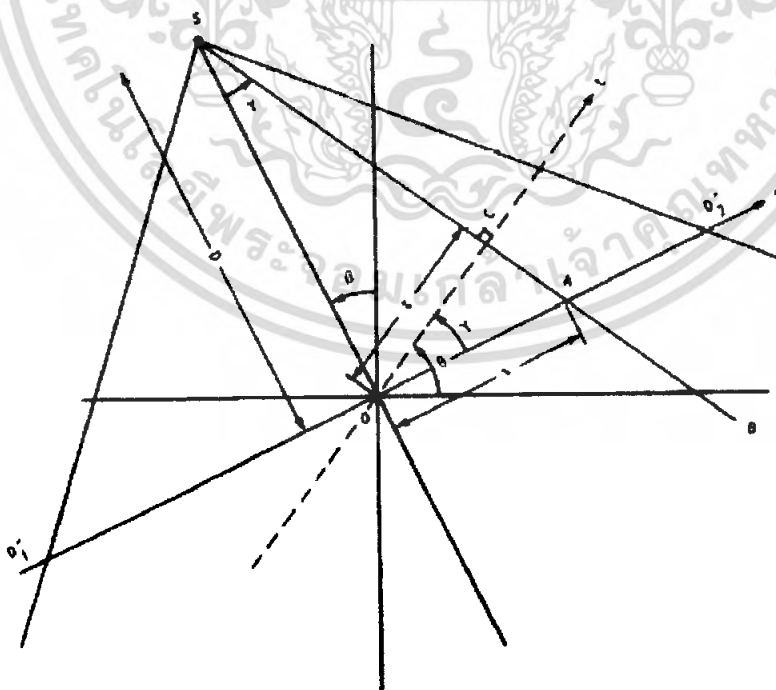
2. ระยะระหว่างเส้นแสงเท่ากัน (Equally space collinear detectors)

กำหนดให้ $R_\beta(s)$ เป็นโปรเจกชันแบบพัด (fan-beam projection) โดยที่มุมภายในระหว่างเส้นแสงห่างเท่าๆกัน จากรูปที่ 2.48 ให้ $\overline{D_1 D_2}$ เป็นแกนคิเทกเตอร์เสมือนตัดผ่านจุดหมุนหรือจุดกำเนิด ถ้าพิจารณาเส้นแสงจากแหล่งกำเนิดแสงไปยังฉากรับ \overline{SB} ซึ่งผ่านจุด A บนแกน $\overline{D_1 D_2}$ เช่นเดียวกับจุด B บน $\overline{D_1 D_2}$ จากรูปที่ 3.3 แกนคิเทกเตอร์สมมติ $\overline{D_1 D_2}$ พิจารณาเส้นแสง \overline{SA} พบว่าค่า s มีระยะเท่ากับ \overline{OA} และถ้าสมมติการสร้างข้อมูลโปรเจกชันพิจารณาทางเดินแสงเป็นแบบขนานเส้นแสง \overline{SA} ก็คือเส้นแสงตำแหน่งที่ t ทำมุม θ หรือ $p_\theta(t)$ ดังนั้นเขียนสมการความสัมพันธ์ระหว่าง β กับ t สำหรับการพิจารณาทางเดินแสงแบบขนาน (parallel-beam projection) ดังสมการที่ (2.21)

$$\begin{aligned} t &= s \cos \gamma & \theta &= \beta + \gamma \\ t &= \frac{sD}{\sqrt{D^2 + s^2}} & \theta &= \beta + \tan^{-1} \frac{s}{D} \end{aligned} \quad 2.21$$



รูปที่ 2.48 ข้อมูลโปรเจกชันเมื่อพิจารณาทางเดินแสงแบบพืดที่มีระยะห่างระหว่างหัวรับเท่ากัน
(Equally space collinear detectors)



รูปที่ 2.49 เวกาณิตของทางเดินแสงแบบพืดที่พิจารณาระยะห่างระหว่างหัวรับเท่ากัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยที่มุม AOC เท่ากันกับมุม OSC และ D คือระยะห่างระหว่างแหล่งกำเนิดแสงที่จุด S ถึงจุดกำเนิด O จากสมการโปรเจกชันแบบขนานเขียนสมการการสร้างภาพตัดขวางแบบฟิลเตอร์แบ็คโปรเจกชันได้คือ

$$f(r, \phi) = \frac{1}{2} \int_0^{2\pi} \int_{-t_m}^{t_m} p_\theta(t) h(r \cos(\theta - \phi) - t) dt d\theta \quad (2.22)$$

เมื่อ $f(r, \phi)$ คือภาพที่สร้างได้ในระบบพิกัดทรงกลม (Polar coordinates) ให้แทนความสัมพันธ์สมการ (2.21) ในสมการ (2.22) จะได้

$$f(r, \phi) = \frac{1}{2} \int_{-\tan(s_m/D)}^{2\pi - \tan(s_m/D)} \int_{-s_m}^{s_m} p_{\beta+\gamma} \left(\frac{sD}{\sqrt{D^2 + s^2}} \right) \cdot h \left(r \cos \left(\beta + \tan^{-1} \left(\frac{s}{D} \right) - \phi \right) - \frac{sD}{\sqrt{D^2 + s^2}} \right) \frac{D^3}{(D^2 + s^2)^{3/2}} ds d\beta \quad (2.23)$$

โดยที่

$$dt d\theta = \frac{D^3}{(D^2 + s^2)^{3/2}} ds d\beta \quad (2.24)$$

จากสมการที่ (2.23) s_m คือค่าที่มากที่สุดของ s ของแต่ละโปรเจกชันและเท่ากับ t_m ของโปรเจกชันแบบขนาน $-\tan(s_m/D)$ และ $2\pi - \tan^{-1}(s_m/D)$ เป็นมุมอยู่ในช่วง 0 ถึง 360 องศาเขียนความสัมพันธ์ระหว่างข้อมูลโปรเจกชันแบบตัดกับแบบขนานได้เป็น

$$R_\beta(s) = p_{\beta+\gamma} \left(\frac{sD}{\sqrt{D^2 + s^2}} \right) \quad (2.25)$$

แทน $R_\beta(s)$ โปรเจกชันแบบตัดในสมการ (3.23) เขียนสมการได้เป็น

$$f(r, \phi) = \frac{1}{2} \int_{-\tan(s_m/D)}^{2\pi - \tan(s_m/D)} \int_{-s_m}^{s_m} R_\beta(s) \cdot h \left(r \cos \left(\beta + \tan^{-1} \left(\frac{s}{D} \right) - \phi \right) - \frac{sD}{\sqrt{D^2 + s^2}} \right) \frac{D^3}{(D^2 + s^2)^{3/2}} ds d\beta \quad (2.26)$$

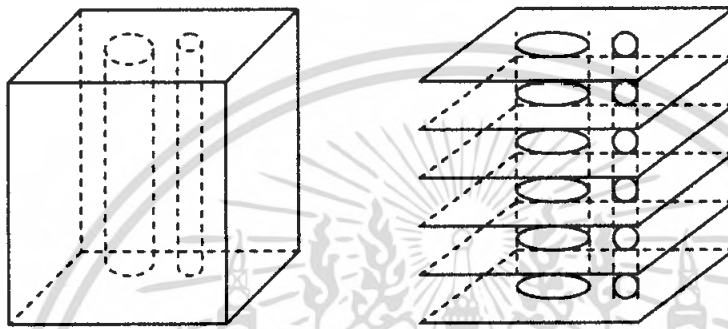
เทอมของฟิลเตอร์ h เขียนใหม่ได้เป็น

$$r \cos \left(\beta + \tan^{-1} \left(\frac{s}{D} \right) - \phi \right) - \frac{sD}{\sqrt{D^2 + s^2}} = r \cos(\beta - \phi) \frac{D}{\sqrt{D^2 + s^2}} - (D + r \sin(\beta - \phi)) \frac{s}{\sqrt{D^2 + s^2}} \quad (2.27)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

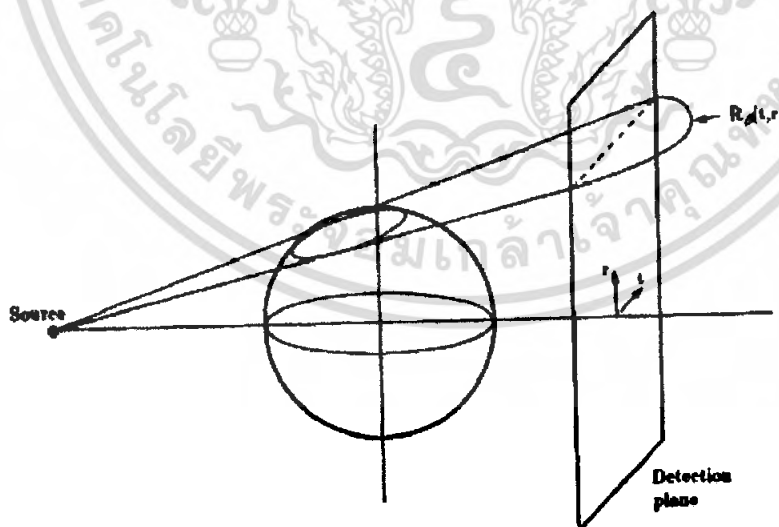
2.3.2 การสร้างภาพตัดขวางแบบกรวย (Cone-beam reconstruction)

การสร้างภาพจากวัตถุที่เป็น 3 มิติจากวิธีการสร้างภาพตัดขวางโดยพิจารณาทางเดินแสงแบบขนานและแบบพัดนั้นสามารถทำได้โดยการสร้างภาพตัดขวางของวัตถุแบบ 3 มิติทีละสไลด์หรือทีละ 2 มิติ แล้วค่อยนำภาพตัดขวางแต่ละสไลด์มารวมกันดังรูปที่ 2.50



รูปที่ 2.50 การสร้างภาพตัดขวางจากวัตถุ 3 มิติ

ในการสร้างภาพตัดขวางที่พิจารณาทางเดินแสงแบบกรวย ใช้การพิจารณาเช่นเดียวกับทางเดินแสงแบบพัด คือแทนทางเดินแสงแบบพัดที่ทำมุมเอียงกับระนาบ (x,y) ของภาพแต่ละสไลด์มีจุดหมุนอยู่ที่ตำแหน่งแหล่งกำเนิดแสงดังรูป 2.51



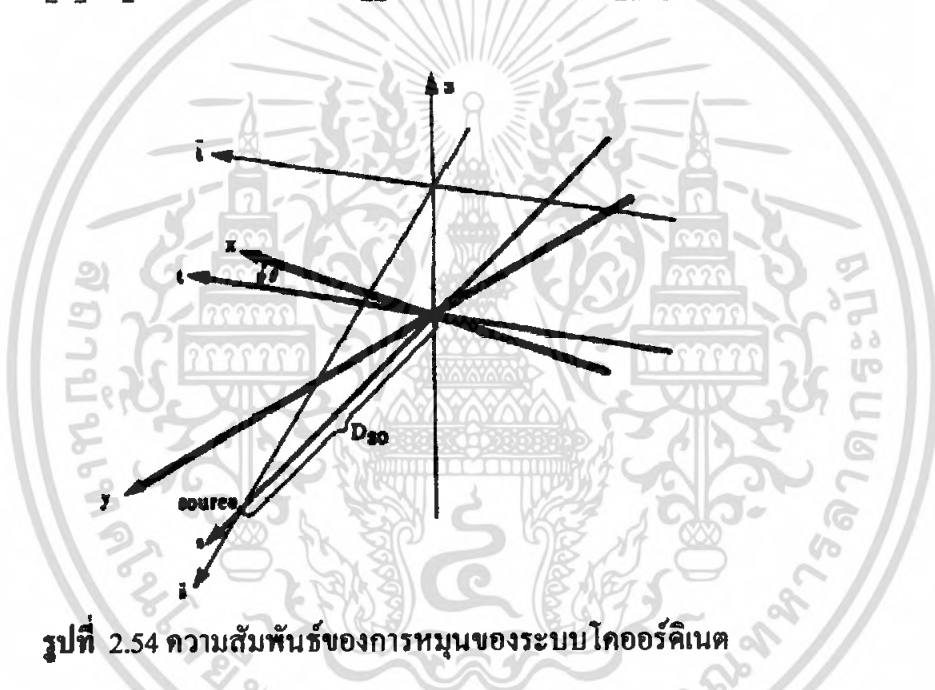
รูปที่ 2.53 การพิจารณาทางเดินแสงแบบกรวยในรูปแบบทางเดินแสงแบบพัด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.3 โปรเจกชันของวัตถุ 3 มิติ(Three-Dimension Projection)

โปรเจกชันของวัตถุ 3 มิติประกอบด้วย 2 ส่วนคือ โคออร์ดิเนตของทางเดินแสง(t, s, r) และโคออร์ดิเนตของวัตถุ (x, y, z) มีความสัมพันธ์กันจากการหมุนดังรูปที่ 3.8 (coordinate-rotation) ที่มี 2 การหมุนคือรอบแกน z ของลำดับมุมการฉายแสงเท่ากับ θ และที่จุดกำเนิดแสงรอบแกน r เป็นมุมเท่ากับ γ ซึ่งเขียนสมการความสัมพันธ์ในฟังก์ชันของเมตริก [3,14] คือ

$$\begin{bmatrix} t \\ s \\ r \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \gamma & \sin \gamma \\ 0 & -\sin \gamma & \cos \gamma \end{bmatrix} \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad (2.28)$$



รูปที่ 2.54 ความสัมพันธ์ของการหมุนของระบบโคออร์ดิเนต

โปรเจกชันแบบขนานของวัตถุ 3 มิติ f มีสมการคือ

$$P_{\theta, \gamma}(t, r) = \int_{-s_m}^{s_m} f(t, s, r) ds \quad (2.29)$$

จากสมการที่ (2.29) โปรเจกชันเป็นฟังก์ชัน 4 ตัวแปรคือ (t, θ) เป็นตัวแปรสำหรับระนาบขนาดและทิศทางของระบบพิกัด (x, y) และ (r, γ) สำหรับระบบพิกัด (s, z) แต่ในระบบทางเดินแสงแบบกรวย (cone-beam system) แหล่งกำเนิดแสงจะถูกหมุนด้วยมุมเท่ากับ β และผลรวมของแสงที่วัดได้แทนด้วย $R_\beta(p', \zeta')$ โดยที่

$$p = \frac{p' D_{SO}}{D_{SO} + D_{DE}} \quad \zeta = \frac{\zeta' D_{SO}}{D_{SO} + D_{DE}} \quad (2.30)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อ D_{so} คือระยะห่างระหว่างจุดหมุน (Center of rotation) ถึงแหล่งกำเนิดแสง และ D_{DE} คือระยะห่างระหว่างจุดหมุนถึงฉากรับซึ่งเขียนความสัมพันธ์ของตัวแปรในระบบทางเดินแสงทั้งสองได้คือ

$$t = p \frac{D_{so}}{\sqrt{D_{so}^2 + \zeta^2}} \quad (2.31)$$

$$\theta = \beta + \tan^{-1} \left[\frac{p}{D_{so}} \right] \quad (2.32)$$

$$r = \zeta \frac{D_{so}}{\sqrt{D_{so}^2 + \zeta^2}} \quad (2.33)$$

โดยที่ r และ ζ ทำหน้าที่ระบุตำแหน่งของระนาบแสงแบบพัด (Tilted fan) ที่ผ่านวัตถุ

2.3.4 FeldKamp หรือ ฟิลเตอร์แบ็คโปรเจกชันแบบ 3 มิติ (3D Filtered Backprojection)

การสร้างภาพกลับวิธีฟิลเตอร์แบ็คโปรเจกชันแบบ 3 มิติ นั้นมีวิธีการเช่นเดียวกับกับวิธีฟิลเตอร์แบ็คโปรเจกชันแบบขนาน แต่ต่างกันที่เส้นทางเดินของแสง ซึ่งสามารถพิจารณาทางเดินแสงให้เป็นรูปพัด (Fan-beam) ในระนาบ 2 มิติ แล้วนำระนาบดังกล่าวหมุนรอบแกน p โดยจุดหมุนอยู่ที่แหล่งกำเนิดแสง

เริ่มแรกให้พิจารณาสมการการสร้างภาพกลับแบบพัด (Fan-beam reconstruction) ในระบบพิกัด (r, ϕ) คือ

$$g(r, \phi) = \frac{1}{2} \int_0^{2\pi} \frac{1}{U^2} \int_{-\infty}^{\infty} R_{\beta}(p) h(p'-p) \frac{D_{so}}{\sqrt{D_{so}^2 + p^2}} dp d\beta \quad (2.34)$$

$$p' = \frac{D_{so} r \cos(\beta - \phi)}{D_{so} + r \sin(\beta - \phi)} \quad h(p) = \int_{-w}^w |\omega| \cdot e^{j\omega p} d\omega \quad (2.35)$$

$$U(r, \phi, \beta) = \frac{D_{so} + r \sin(\beta - \phi)}{D_{so}} \quad (2.36)$$

เขียนสมการความสัมพันธ์ (r, ϕ) ให้อยู่ในระบบโคออร์ดิเนตการหมุน (t, s) ได้คือ

$$t = x \cos \beta + y \sin \beta \quad s = y \cos \beta - x \sin \beta \quad (2.37)$$

$$x = r \cos \phi \quad y = r \sin \phi \quad (2.38)$$

$$p' = \frac{D_{so}}{D_{so} - s} \quad U(x, y, z) = \frac{D_{so} - s}{D_{so}} \quad (2.39)$$

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์การใช้งานเพื่อการศึกษาค้นคว้าเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

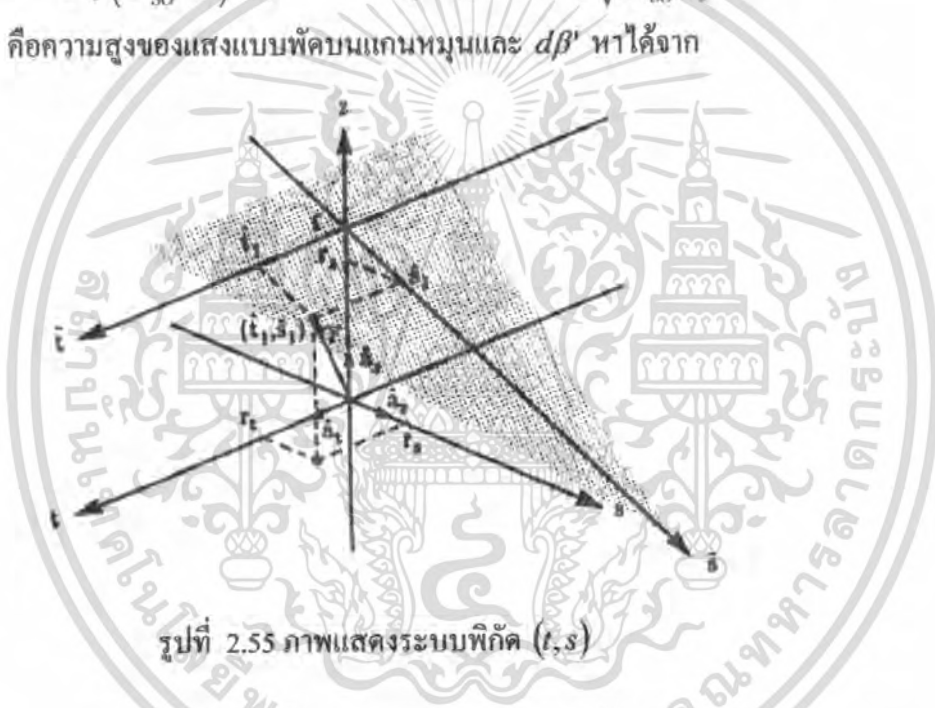
สมการสร้างภาพแบบพัดในระบบโคออร์ดิเนต (t, s) เขียนได้เป็น

$$g(t, s) = \frac{1}{2} \int_0^{2\pi} \frac{D_{SO}^2}{(D_{SO} - s)^2} \int_{-\infty}^{\infty} R_{\beta}(p, \zeta) h\left(\frac{D_{SO} \cdot t}{D_{SO} - s} - p\right) \frac{D_{SO}}{\sqrt{D_{SO}^2 + p^2}} dp d\beta \quad (2.40)$$

เมื่อแสงเปลี่ยนระนาบไปดังรูปที่ 2.55 ทำให้ไม่สามารถใช้สมการที่ (2.40) ได้เนื่องจาก
ระยะทางของแสงเปลี่ยนไป เขียนสมการสร้างภาพแบบพัดสำหรับ (t', s') คือ

$$g(t', s') = \frac{1}{2} \int_0^{2\pi} \frac{D'_{SO}{}^2}{(D'_{SO} - s')^2} \int_{-\infty}^{\infty} R_{\beta'}(p, \zeta) h\left(\frac{D'_{SO} \cdot t'}{D'_{SO} - s'} - p\right) \frac{D'_{SO}}{\sqrt{D'_{SO}{}^2 + p^2}} dp d\beta' \quad (2.41)$$

โดยที่ ζ คือความสูงของแสงแบบพัดบนแกนหมุนและ $d\beta'$ หาได้จาก



รูปที่ 2.55 ภาพแสดงระบบพิกัด (t, s)

$$D_{SO} d\beta = D'_{SO} d\beta' \quad d\beta' = \frac{D_{SO} d\beta}{D'_{SO}} \quad (2.42)$$

$$t = t' \quad s' = D'_{SO} \frac{s}{D_{SO}} \quad \zeta = D_{SO} \frac{z}{D_{SO} - s} \quad (2.43)$$

$$D'_{SO} = \sqrt{D_{SO}^2 + \zeta^2} \quad (2.44)$$

แทนค่า D'_{SO} ด้วย D_{SO} และ β' ด้วย β เขียนสมการ 2.41) สำหรับทุกระนาบแสงได้เป็น

$$g(t, s) = \frac{1}{2} \int_0^{2\pi} \frac{D_{SO}^2}{(D_{SO} - s)^2} \int_{-\infty}^{\infty} R_{\beta}(p, \zeta) h\left(\frac{D_{SO} \cdot t}{D_{SO} - s} - p\right) \frac{D_{SO}}{\sqrt{D_{SO}^2 + \zeta^2 + p^2}} dp d\beta \quad (2.45)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยขั้นตอนของพีลเตอร์แบ็คโปรเจกชันแบบ 3 มิติดังต่อไปนี้

1. นำโปรเจกชัน $R_\beta(p, \zeta)$ คูณด้วยฟังก์ชัน $\frac{D_{so}}{\sqrt{D_{so}^2 + \zeta^2 + p^2}}$ เพื่อหาค่า คือ

$$R'(p, \zeta, \beta) = R(p, \zeta, \beta) \frac{D_{so}}{\sqrt{D_{so}^2 + \zeta^2 + p^2}} \quad (2.46)$$

2. คอนโวลูชันโปรเจกชัน $R'_\beta(p, \zeta)$ กับ $\frac{1}{2}h(p)$ ได้ค่า $Q_\beta(p, \zeta)$ คือ

$$Q(p, \zeta, \beta) = R'(p, \zeta, \beta) * \frac{1}{2}h(p) \quad (2.47)$$

3. แบ็คโปรเจกชันด้วยค่าแฟกเตอร์นำหนักตามระยะทางของเส้นแสงสำหรับระบบโคออร์ดิเนตแบบ 3 มิติ

$$g(x, y, z) = \int_0^{2\pi} \frac{D_{so}^2}{(D_{so} - s)^2} Q_\beta(p, \zeta, \beta) d\beta \quad (2.48)$$

หรือ

$$f(x, y, z) = \frac{1}{2} \int_0^{2\pi} \frac{D_{so}^2}{(D_{so} - s)^2} \int_{-\infty}^{\infty} R(p, \zeta, \beta) h\left(\frac{D_{so} \cdot t}{D_{so} - s} - p\right) \frac{D_{so}}{\sqrt{D_{so}^2 + \zeta^2 + p^2}} dp d\beta \quad (2.49)$$

2.3.5 การแปลงเชิงเรขาคณิต

การแปลงเชิงเรขาคณิตแบบเชิงเส้น

ในงานทางด้านคอมพิวเตอร์กราฟิกนั้น บางครั้งจำเป็นต้องมีการเคลื่อนย้ายภาพไปยังตำแหน่งใหม่ ย่อ-ขยายภาพ หรือมีการหมุนภาพให้อยู่ในมุมมองที่เหมาะสม ซึ่งสามารถทำได้โดยใช้การแปลงเชิงเรขาคณิต(Geometric transformation)

ในการพิจารณาการแปลงนั้นจะพิจารณาได้ว่าเป็นการแปลงจุดในปริภูมิสามมิติ และเพื่อความเข้าใจง่ายจึงมักจะเขียนอยู่ในรูปของเมทริกซ์ โดยกำหนดให้จุดที่ต้องการแปลงเป็นจุด P มีพิกัดเป็น (x, y, z) ใดๆ เมื่อต้องการย้ายตำแหน่งของจุด P ไปยังพิกัดใหม่คือ P' ซึ่งมีพิกัด (x', y', z') ก็ทำได้โดยการนำพิกัดของจุด P ไปบวกกับเมทริกซ์การแปลงซึ่งก็คือ T สามารถแสดงเมทริกซ์ของการแปลงได้ดังนี้

$$P' = P + T = [x \ y \ z] + [T_x \ T_y \ T_z] \quad (2.50)$$

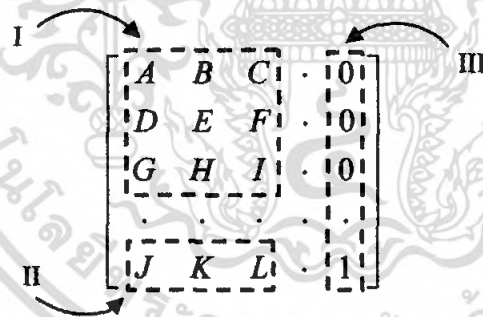
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นอกจากนี้เมื่อนำพิกัดของจุด P มาคูณกับเมทริกซ์การแปลง T ให้ได้พิกัดใหม่คือ P' นั้นจะเป็น การสเกล การหมุน การสะท้อน หรืออื่นๆ ขึ้นอยู่กับค่าพารามิเตอร์ภายในเมทริกซ์การแปลงนั้นและ สามารถเขียนในรูปของเมทริกซ์ได้ดังนี้

$$P' = P \cdot T = [x \ y \ z] \cdot \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} \quad (2.51)$$

สำหรับการแปลงเชิงเรขาคณิตในทางคอมพิวเตอร์กราฟิกโดยทั่วไปแล้วจะประกอบด้วยการ แปลงหลายๆชนิดประกอบกัน และเนื่องจากการคำนวณในรูปของเมทริกซ์จึงทำให้ยากในการ คำนวณเมทริกซ์ที่มีรูปแบบแตกต่างกัน จึงได้มีการนำระบบพิกัดโฮโมจีนียส (Homogeneous Coordinate System) เข้ามาใช้เพื่อแก้ปัญหา

ในระบบพิกัดโฮโมจีนียส จุดในระบบสามมิติ (x, y, z) จะถูกแทนด้วยระบบสี่มิติ นั่นคือ (x, y, z, H) โดยจะมี H เป็นค่าแฟกเตอร์ที่มีหน้าที่สเกลซึ่งจะมีค่าไม่เท่ากับศูนย์ จุด (x, y, z, H) จะถูกนอร์มอลไลซ์ (Normalize) เป็น $(x/H, y/H, z/H, 1)$ โดยทั่วไปแล้วค่า H จะมีค่าเท่ากับ 1 ดังนั้นจุด (x, y, z) ใดๆจะมีค่าในระบบพิกัดโฮโมจีนียสเป็น $(x, y, z, 1)$



รูปที่ 2.56 ส่วนประกอบของระบบพิกัดโฮโมจีนียส

จากรูปที่ 2.56 เมทริกซ์ย่อย I เป็นส่วนที่กำหนดว่าเป็นการแปลงแบบใด เช่น การหมุน การ สเกล การสะท้อน เป็นต้น ส่วนเมทริกซ์ย่อย II เป็นส่วนของการย้ายตำแหน่งแบบเชิงเส้น และใน ส่วนสุดท้ายเมทริกซ์ย่อย III ใช้ในการรวมระหว่งการเคลื่อนย้ายพิกัดและการแปลงที่อยู่ในรูปการ คูณ โดยมีรายละเอียดของการแปลงแบบต่างๆดังนี้

1. การเลื่อนพิกัด (Translation)

เมทริกซ์การแปลง (T_T) ที่ใช้สำหรับการแปลงที่ต้องการเลื่อนพิกัดมีลักษณะดังนี้
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อผู้ดูแลเนื้อหาไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$T_T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ T_x & T_y & T_z & 1 \end{bmatrix} \quad (2.52)$$

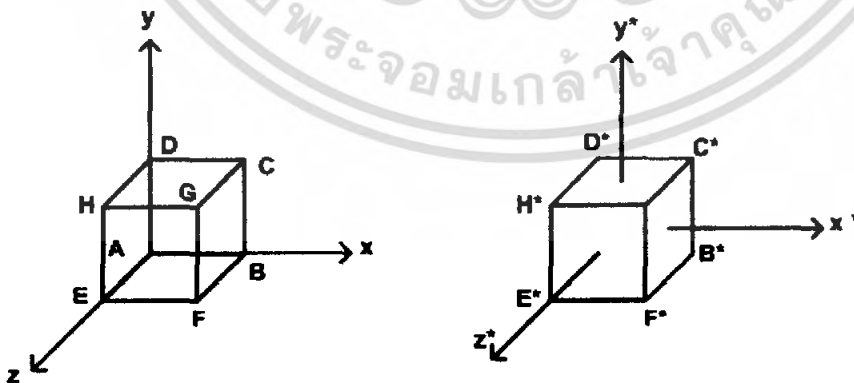
โดยที่ค่า T_x, T_y และ T_z แสดงระยะทางที่ต้องการเคลื่อนย้ายตำแหน่งไปในแนวแกน X, Y และ Z ตามลำดับ และพิกัดของจุดในระบบโฮโมจีเนียส (x', y', z', h) เป็น

(2.53)

หรือ

$$[x' \ y' \ z' \ h] = [x + T_x \ y + T_y \ z + T_z \ 1] \quad (2.54)$$

การเลื่อนเมทริกซ์เป็นการแปลงที่มีประโยชน์มากในทางคอมพิวเตอร์กราฟิก เนื่องจากข้อมูลภาพต้นฉบับที่นำมาใช้จะมีตำแหน่งอยู่ในด้านบวกเท่านั้น จึงจำเป็นต้องการอาศัยการแปลงแบบนี้เพื่อเลื่อนพิกัดกึ่งกลางของข้อมูลต้นฉบับให้มาอยู่ที่จุดกำเนิด $(0,0,0)$ เสียก่อน แสดงในรูปที่ 2.56 จึงถือได้ว่าการแปลงแบบเลื่อนพิกัดเป็นพื้นฐานที่สำคัญของการสร้างภาพทางคอมพิวเตอร์กราฟิก



รูปที่ 2.56 การเลื่อนจุดกึ่งกลางของปริมาตรต้นฉบับมายังจุดกำเนิด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. การสเกล (Scaling)

การสเกลเป็นการแปลงเพื่อปรับเปลี่ยนขนาดของวัตถุที่ต้องการ มีเมทริกซ์การแปลง (S)

ดังนี้

$$S = \begin{bmatrix} S_x & 0 & 0 & 0 \\ 0 & S_y & 0 & 0 \\ 0 & 0 & S_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.55)$$

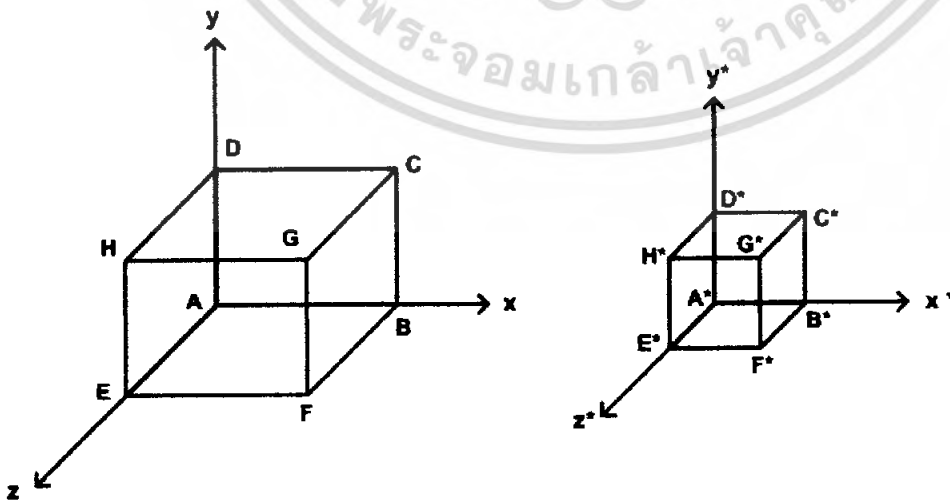
โดยที่ S_x, S_y และ S_z เป็นสัมประสิทธิ์การสเกลสำหรับพิกัด X, Y และ Z ตามลำดับ และพิกัดของจุดในระบบโฮโมจีเนียส (x', y', z', h) เป็น

$$\begin{bmatrix} x' & y' & z' & h \end{bmatrix} = \begin{bmatrix} x & y & z & 1 \end{bmatrix} \cdot \begin{bmatrix} S_x & 0 & 0 & 0 \\ 0 & S_y & 0 & 0 \\ 0 & 0 & S_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.56)$$

หรือ

$$\begin{bmatrix} x' & y' & z' & h \end{bmatrix} = \begin{bmatrix} xS_x & yS_y & zS_z & 1 \end{bmatrix} \quad (2.57)$$

ค่า S_x, S_y และ S_z จะเป็นตัวกำหนดว่าเป็นการย่อหรือขยายรูป กล่าวคือ ถ้า S มีค่ามากกว่า 1 จะเป็นการขยายภาพและถ้า S มีค่าน้อยกว่า 1 จะเป็นการย่อขนาดภาพ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับรูปที่ 2.57 การสเกลวัตถุเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. การหมุนวัตถุ (Rotation)

การหมุนวัตถุเป็นการหมุนวัตถุรอบแกนต่างๆทั้งสามแกน ซึ่งได้แก่ แกน X, Y หรือ Z โดยการอ้างอิงการมองในลักษณะมองออกจากจุดกำเนิดไปตามแนวแกน เมทริกซ์การแปลงของการหมุนรอบแกนต่างๆในทิศทางตามเข็มนาฬิกา มีดังนี้

เมทริกซ์การแปลงของการหมุนรอบแกน X ไปเป็นมุม θ คือ

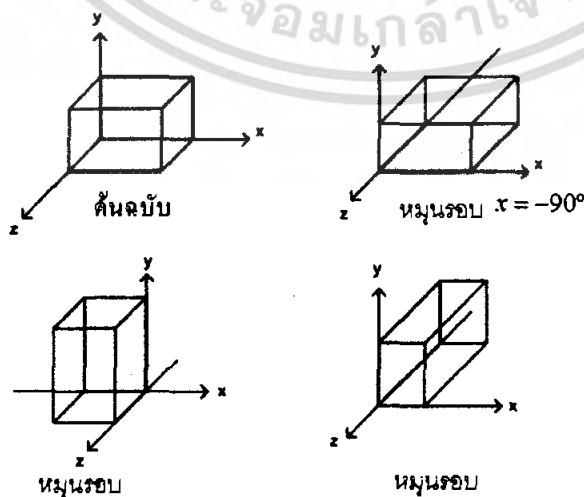
$$R_x = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \theta & \sin \theta & 0 \\ 0 & -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.58)$$

เมทริกซ์การแปลงของการหมุนรอบแกน Y ไปเป็นมุม ϕ คือ

$$R_y = \begin{bmatrix} \cos \phi & 0 & -\sin \phi & 0 \\ 0 & 1 & 0 & 0 \\ \sin \phi & 0 & \cos \phi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.59)$$

เมทริกซ์การแปลงของการหมุนรอบแกน Z ไปเป็นมุม φ คือ

$$R_z = \begin{bmatrix} \cos \varphi & \sin \varphi & 0 & 0 \\ -\sin \varphi & \cos \varphi & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.60)$$



รูปที่ 2.58 การแปลงแบบหมุนรอบแกนทั้งสาม

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้สำหรับการใช้ในเพื่อวัตถุประสงค์เท่านั้น เมื่ออนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. การสะท้อน (Reflection)

การสะท้อนวัตถุเป็นการสะท้อนพิคคของวัตถุ ไปยังพิคคที่อยู่อีกด้านหนึ่งของระนาบการสะท้อน XY, YZ หรือ XZ เมทริกซ์ของการสะท้อนมีดังต่อไปนี้

เมทริกซ์ของการสะท้อนกับระนาบ XY คือ

$$Rf_{XY} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.61)$$

จะเปลี่ยนเฉพาะพิคค z โดย $z' = -z$

เมทริกซ์ของการสะท้อนกับระนาบ YZ คือ

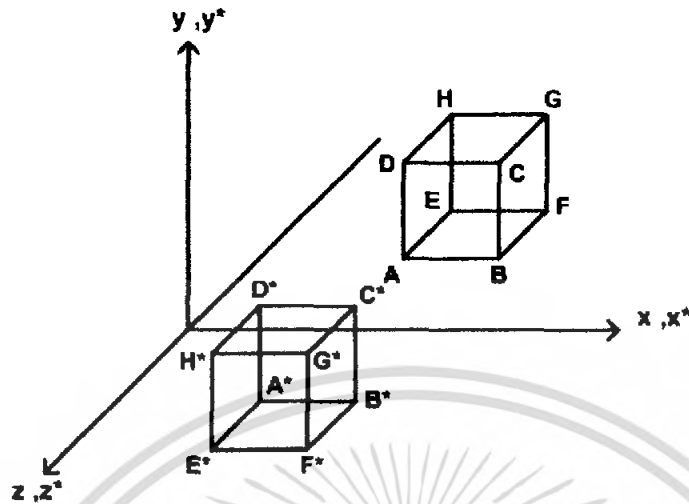
$$Rf_{YZ} = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.62)$$

จะเปลี่ยนเฉพาะพิคค x โดย $x' = -x$

เมทริกซ์ของการสะท้อนกับระนาบ XZ คือ

$$Rf_{XZ} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.63)$$

จะเปลี่ยนเฉพาะพิคค y โดย $y' = -y$



รูปที่ 2.59 การสะท้อนวัตถุกับระนาบ XY

5. การเฉือน (Shearing)

การเฉือนในปริภูมิสามมิติเป็นการแปลงพิกัดตามแกนของสองแกนใดๆ (สมมติว่าเป็น x และ y) โดยสัมพันธ์กับค่าของพิกัดตามแนวแกนที่สาม (z) เป็นผลให้ค่าพิกัดตามแนวแกนของสองแกนแรกเปลี่ยนไป ในขณะที่ค่าพิกัดตามแนวแกนที่สามนั้นมีค่าคงเดิม เมทริกซ์ของการเฉือนโดยสัมพันธ์กับแกน X

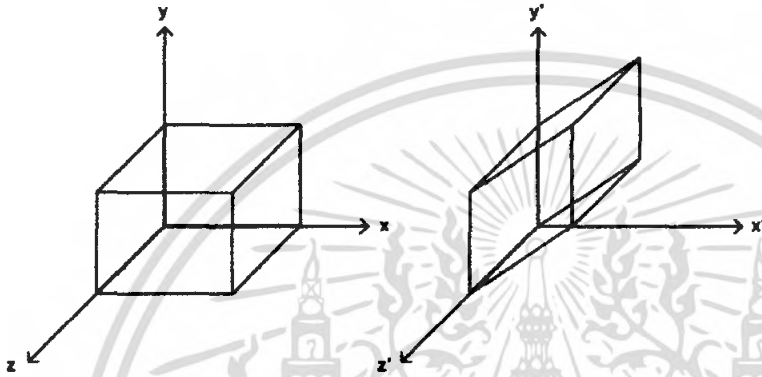
$$Sh_x = \begin{bmatrix} 1 & Sh_{xy} & Sh_{xz} & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.64)$$

เมทริกซ์ของการเฉือนโดยสัมพันธ์กับแกน Y

$$Sh_y = \begin{bmatrix} 1 & 0 & 0 & 0 \\ Sh_{yx} & 1 & Sh_{yz} & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.65)$$

เมทริกซ์ของการเฉือนโดยสัมพันธ์กับแกน Z

$$Sh_z = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ Sh_{zx} & Sh_{zy} & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.66)$$



รูปที่ 2.60 การเฉือนโดยสัมพันธ์กับค่า z เมื่อ $Sh_{zx} = 0$ และ $Sh_{zy} = 1$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.6 การสร้างภาพสามมิติโดยใช้ OpenGL

OpenGL เป็นซอฟต์แวร์ไลบรารีที่ใช้ติดต่อกับฮาร์ดแวร์แสดงผล โดยมีคำสั่งพื้นฐานประมาณ 120 คำสั่งที่สามารถใช้กำหนดคุณลักษณะและควบคุมการทำงานของแอปพลิเคชันสามมิติ OpenGL ถูกออกแบบมาโดยไม่ยึดติดกับระบบ สามารถทำงานได้บนทุกๆแพลตฟอร์ม (Portability) เพื่อที่จะบรรลุเป้าหมายนี้ OpenGL จะไม่มีคำสั่งที่จัดการกับระบบปฏิบัติการเลย อีกทั้งยังไม่มีคำสั่งเพื่อรับอินพุตจากผู้ใช้อีกด้วย สิ่งที่ OpenGL เตรียมไว้ให้สำหรับสร้างรูปจำลองสามมิติคือรูปทรงเรขาคณิตอย่างง่ายได้แก่ จุด เส้น และรูปหลายเหลี่ยมซึ่งผู้ใช้งานจะต้องนำรูปทรงเหล่านี้มาประกอบกันเพื่อให้เกิดรูปทรงสามมิติที่ซับซ้อน

คำสั่งและชนิดของข้อมูลของ OpenGL

การใช้งานคำสั่ง OpenGL จะต้อง include ไฟล์ header ชื่อ gl.h คำสั่งของ OpenGL จะขึ้นต้นด้วย gl อย่างเช่น glColor() คำสั่งที่จะขึ้นต้นด้วย GL_ เช่น GL_COLOR_BUFFER_BIT สำหรับตัวลงท้ายหรือ suffix ของบางคำสั่งจะประกอบด้วยตัวเลขและตัวอักษรเช่น glColor3f ซึ่งเลข 3 บ่งบอกถึงจำนวนตัวแปรอินพุตคือสี RGB หากเป็น glColor4f จะหมายถึง RGBA ส่วน f หมายถึงอินพุตจะต้องเป็นข้อมูลชนิดเลขทศนิยม บางคำสั่งอาจรับข้อมูลอินพุตที่แตกต่างกันได้ถึง 8 ชนิด

การทำงานของ OpenGL จะเป็นลักษณะ State Machine คือเมื่อสั่งให้โปรแกรมอยู่ในสถานะใดสถานะนั้นจะคงอยู่ตลอดจนกว่าจะมีการเปลี่ยนแปลง อย่างเช่นการตั้งค่าสีซึ่งเป็น State Variable เมื่อตั้งค่าสีไปครั้งหนึ่ง วัตถุที่ดูกวาดหลังจากนั้นจะมีสีนั้นตลอดจนกว่าจะมีคำสั่งเปลี่ยนสี

GLU: OpenGL Utility Library

GLU เป็นการเอาชุดคำสั่งของ OpenGL ที่ถูกใช้บ่อยๆมารวมกันเป็นคำสั่งใหม่ซึ่งทำให้การเขียนโปรแกรมสะดวกขึ้นอย่างเช่นการทำ Mipmapping สำหรับ Texture Mapping, การแปลงโคออดิเนตแบบ Orthogonal และ Perspective, การแบ่งข่อยรูปเหลี่ยมหรือ Polygon Tessellation, การเรนเดอร์รูปทรงพื้นฐานเช่นทรงกลม ทรงกระบอก และแผ่นจาน, และการวาดเส้นโค้งและพื้นผิวโค้งโดยใช้ Non-Uniform Rational B-Spline (NURBS) คำสั่งของ GLU จะขึ้นต้นด้วยคำว่า glu เสมอ การเรียกใช้จะต้อง include ไฟล์ header ชื่อ glu.h

GLUT: OpenGL Utility Toolkit

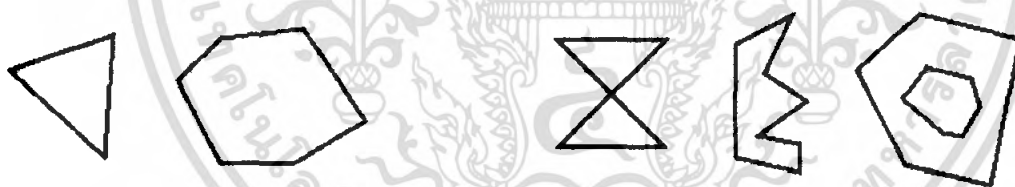
GLUT เขียนขึ้นโดย Mark Kilgard เป็นเครื่องมือช่วยติดต่อกันระหว่างแอปพลิเคชันของ OpenGL กับระบบปฏิบัติการแบบ Windows หรือเรียกว่าเป็น API (Application Programming Interface) ทำให้การเขียนโปรแกรมโดยใช้ OpenGL มีความซับซ้อนน้อยลง เหมาะสำหรับการพัฒนาโปรแกรมขนาดเล็กถึงขนาดกลาง คำสั่งของ GLUT จะขึ้นต้นด้วยคำว่า glut

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

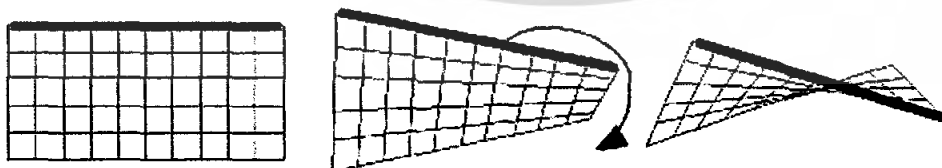
รูปทรงพื้นฐาน (Geometric Primitives)

รูปทรงพื้นฐานของ OpenGL ได้แก่ จุด, เส้น, และรูปเหลี่ยม โดยรูปทรงทั้งสามนี้ถูกกำหนดตำแหน่งโดยคู่ลำดับสามมิติ (x, y, z) หรือคู่ลำดับโฮโมจีนีอัส (x, y, z, w) (โดยปกติแล้ว $w = 1$) ที่เรียกว่า vertex ซึ่งใช้ 1 vertex สำหรับจุด 2 vertex สำหรับเส้น และมากกว่า 2 vertex สำหรับรูปเหลี่ยม OpenGL ได้กำหนดลักษณะของรูปเหลี่ยมที่เหมาะสมโดยใช้ข้อบังคับสามข้อคือ ขอบของรูปเหลี่ยมจะต้องไม่ตัดกันเอง รูปทรงของรูปเหลี่ยมจะต้องไม่แหงงเว้าเข้าไป และจะต้องไม่มีภายในรูปเหลี่ยม (รูปร่างที่ 1 ประกอบ) แต่ไม่มีข้อกำหนดสำหรับจำนวนมุมของรูปเหลี่ยม ข้อบังคับเหล่านี้ทำให้การแสดงผลมีความรวดเร็ว และหากใช้รูปเหลี่ยมที่ไม่เหมาะสมก็อาจทำให้การเรนเดอร์ภาพผิดพลาดได้ อย่างไรก็ตามรูปเหลี่ยมที่ไม่เหมาะสมสามารถแบ่งเป็นรูปเหลี่ยมย่อยๆ ได้โดยผู้ใช้หรือใช้ไลบรารี GLU ซึ่งมีคำสั่งที่ใช้ขอยรูปเหลี่ยมที่ซับซ้อนด้วย (Tessellation)

ข้อควรระวังสำหรับการเรนเดอร์รูปเหลี่ยมคือเมื่อ vertex ของรูปเหลี่ยมไม่ได้วางอยู่บนระนาบเดียวกัน การหมุนและการเปลี่ยนมุมมองอาจทำให้รูปเหลี่ยมนั้นกลายเป็นรูปเหลี่ยมที่ไม่เหมาะสมได้ตัวอย่างในรูปที่ 2 เมื่อรูปสี่เหลี่ยมมี vertex ที่ไม่ได้อยู่บนระนาบเดียวกันโดยวางเหลี่ยมกันเล็กน้อย เมื่อเกิดการหมุนจะทำให้รูปสี่เหลี่ยมนั้นมีลักษณะที่ไม่เหมาะสมคือเว้าเข้าไป ซึ่งจะทำให้การเรนเดอร์ผิดพลาดปัญหานี้สามารถหลีกเลี่ยงได้หากใช้เฉพาะรูปสามเหลี่ยมเพราะจุดทั้งสามจะวางตัวอยู่ในระนาบเดียวกันแน่นอน



รูปที่ 2.61 รูปเหลี่ยมที่เหมาะสมและไม่เหมาะสม



รูปที่ 2.62 รูปเหลี่ยมที่ไม่ได้ระนาบเมื่อหมุนเปลี่ยนมุมมองจะทำให้เกิดส่วนเว้า

เส้นโค้งและพื้นผิวโค้งสำหรับ OpenGL แล้วไม่ถือว่าเป็นรูปทรงพื้นฐาน แต่สามารถประมาณได้โดยใช้เส้นตรงหรือพื้นผิวย่อยๆมาประกอบกัน ดังรูปที่ 2.63 และ 2.64

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.63 เส้นโค้งที่เกิดจากการประกอบกันของเส้นตรง



รูปที่ 2.64 ทรงกลมที่เกิดจากการประกอบกันของพื้นผิวเรียบ

แสง

แสงมีความสำคัญมาก ในการสร้างภาพสามมิติ วัตถุส่วนใหญ่จะไม่ดูเหมือนสามมิติจนกว่าจะมีการให้แสง ดังตัวอย่างทรงกลมในรูปที่ 2.65 จะเห็นว่าทรงกลมที่ยังไม่ให้แสงจะมีลักษณะเหมือนแผ่นจาน 2 มิติ



รูปที่ 2.65 ทรงกลมที่ไม่มีแสงตกกระทบและทรงกลมที่มีแสงตกกระทบ

เมื่อแสงตกกระทบวัตถุ เราจะสามารถรับสีที่แตกต่างกันของวัตถุขึ้นอยู่กับพลังงานของโฟตอนที่ตกกระทบเซลล์ประสาทรูปกรวย โฟตอนเหล่านั้นมาจากแหล่งกำเนิดแสง บางส่วนของโฟตอนจะถูกดูดกลืนและส่วนที่เหลือจะถูกสะท้อน โดยพื้นผิวของวัตถุ พื้นผิวที่แตกต่างกันจะมีคุณสมบัติการสะท้อนแสงที่แตกต่างกัน พื้นผิวที่มันเงาจะสะท้อนแสงไปยังทิศทางที่เจาะจง พื้นผิวหยาบจะสะท้อนแสงให้กระจายไปในทุกทิศทาง แต่โดยปกติแล้วพื้นผิวส่วนใหญ่จะมีคุณสมบัติอยู่ระหว่างพื้นผิวสองแบบนี้

OpenGL จะคำนวณแสงโดยแยกแสงเป็น 3 องค์ประกอบคือแดง เขียวและน้ำเงิน (RGB) ดังนั้นสีของแสงจากแหล่งกำเนิดจะถูกกำหนดโดยปริมาณขององค์ประกอบเหล่านี้ และสีของพื้นผิววัตถุจะถูกกำหนดโดยเปอร์เซ็นต์ของแสงสีแดง เขียวและน้ำเงินที่ตกกระทบและถูกสะท้อนออกไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับกระบวนงานเพื่อการศึกษานี้เท่านั้น ไม่สามารถนำไปใช้ประโยชน์ด้านการค้า
ตัวอย่างเช่นลูกบอลสีแดงจะดูดกลืนแสงสีเขียวและแสงสีน้ำเงินแต่จะสะท้อนแสงสีแดง เมื่ออยู่ใน
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งยังมีให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แสงขาวจะมองเห็นเป็นสีแดง แต่ถ้าอยู่ในแสงสีเขียวมันจะกลายเป็นสีดำ สมการคำนวณแสงของ OpenGL นั้นเป็นแค่การประมาณการแต่ให้ผลลัพธ์ที่ดีในเวลาคำนวณที่สั้น

แหล่งกำเนิดแสงใน OpenGL สามารถกำหนดให้มาได้จากหลายทิศทาง โดยสามารถกำหนดให้มาจากทิศทางที่แน่นอนได้หรือจะให้มันเป็นแสงที่สะท้อนไปมารอบๆจาก ตัวอย่างเช่นเมื่อเราเปิดหลอดไฟในห้อง แสงส่วนใหญ่จะมาจากหลอดไฟนั้นโดยตรง แต่แสงบางส่วนจะสะท้อนไปมาในผนังห้องหลายรอบก่อนที่จะถึงตาเรา แสงที่สะท้อนไปมาในห้องนี้เรียกว่า Ambient Light ซึ่งถูกสมมติว่าไม่ทราบแหล่งกำเนิดที่แน่นอนแต่จะหายไปเมื่อเราปิดหลอดไฟนั้น

โมเดลของแหล่งกำเนิดแสงใน OpenGL จะถูกแบ่งเป็น 4 แบบใหญ่คือ Emitted, Ambient, Diffuse, และ Specular Light โดยโมเดลทั้ง 4 นี้จะถูกกำหนดองค์ประกอบสี RGB ที่แตกต่างกัน และถูกคำนวณแยกกันแล้วนำมารวมกันภายหลัง

Emitted Light – เป็นแสงที่ส่งออกมาจากตัววัตถุเอง

Ambient Light – เป็นแสงที่กระจัดกระจายจากการสะท้อนจนไม่รู้แหล่งที่มา ทำให้ดูเหมือนกับมาจากทุกทิศทาง เมื่อแสงแบบนี้ตกกระทบวัตถุจะสะท้อนเท่าๆกันในทุกทิศทาง แสงจากหลอดไฟแบ็คไลท์ในห้องจะมีองค์ประกอบนี้อยู่มากเนื่องจากถูกสะท้อนหลายครั้งก่อนที่จะมาถึงตา ส่วนแสงจากสปอตไลท์จะมีองค์ประกอบนี้น้อยมากเนื่องจากแสงจะพุ่งไปในทิศทางเดียว

Diffuse Light – เป็นแสงที่มาจากแหล่งกำเนิดโดยตรงแต่เมื่อตกกระทบวัตถุจะสะท้อนออกไปเท่ากันทุกทิศทาง

Specular Light – มาจากแหล่งกำเนิดที่เจาะจงเช่นกันแต่จะสะท้อนไปในทิศทางที่เจาะจง คือมุมสะท้อนเท่ากับมุมตกกระทบ แสงจากเลเซอร์ที่ตกกระทบกระจกที่มีคุณภาพจะให้การสะท้อนเป็น Specular เกือบ 100 เปอร์เซ็นต์ วัตถุที่มันเงาเช่นพลาสติกจะมีคุณสมบัติของ Specular มากกว่าวัตถุที่หยาบเช่นชอล์ก

วัตถุมิโมเดลการสะท้อนแสงสามแบบคือ Ambient, Diffuse, และ Specular Reflectance ที่ถูกกำหนดสีไว้แตกต่างกัน Ambient Reflectance ของวัตถุจะใช้สำหรับคำนวณสีของวัตถุที่แสดงออกมาเมื่อถูก Ambient Light ตกกระทบ เช่นเดียวกับ Diffuse และ Specular Reflectance ซึ่งคำนวณกับ Diffuse และ Specular Light ตามลำดับ โดยปกติแล้วสีของวัตถุที่ถูกกำหนดโดย Ambient และ Diffuse Reflectance จะเหมือนหรือคล้ายคลึงกัน ส่วน Specular Reflectance จะกำหนดเป็นสีขาวหรือเทา ซึ่งทำให้จุดที่สะท้อนเป็นสีของแหล่งกำเนิดแสง อย่างเช่นลูกบอลมันเงาสีแดงเมื่อถูกฉายด้วยแสงสีขาวพื้นที่ส่วนใหญ่จะเป็นสีแดงแต่จะมีส่วนหนึ่งที่มีมุมตกกระทบเท่ากับมุมสะท้อนซึ่งจะเป็นสีขาว

การกำหนดปริมาตรสีสำหรับแสงจะกำหนดเป็นเปอร์เซ็นต์ของความเข้มสูงสุดของแต่ละสี ถ้าค่า RGB เป็น 1.0 ทั้งหมดจะได้แสงสีขาว ถ้าเป็น 0.5 ทั้งหมด จะได้สีขาวที่ความเข้มต่ำลงครึ่งหนึ่ง

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แต่ถ้า $R=G=1.0, B=0.0$ จะเป็นแสงสีเหลือง ส่วนการกำหนดปริมาณสีที่สะท้อนออกมาจากวัตถุจะกำหนดเป็นสัดส่วนของสีที่สะท้อนออกมา อย่างเช่นถ้า $R=1.0, G=0.5, B=0.0$ วัตถุจะสะท้อนแสงสีแดงทั้งหมด สะท้อนแสงสีเขียวเพียงครึ่งหนึ่งของแสงสีเขียวที่ตกกระทบและดูดกลืนแสงสีน้ำเงินทั้งหมด สมมติว่าแสงมีค่าความเข้มเป็น(LR, LG, LB) และวัตถุมีค่าการสะท้อนแสงเป็น (MR, MG, MB) แสงที่เข้าสู่ตาจะเป็น (LR*MR, LG*MG, LB*MB) ถ้าหากมีแสงจากสองแหล่งเข้าสู่ตาเป็น (R1, G1, B1) และ (R2, G2, B2) แสงรวมจะเป็น (R1+R2, G1+G2, B1+B2) หากค่าที่รวมได้มีค่ามากกว่า 1.0 ซึ่งสว่างเกินกว่าที่จอภาพจะแสดงผลได้ ค่านั้นจะถูกบีคลงเป็น 1.0



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

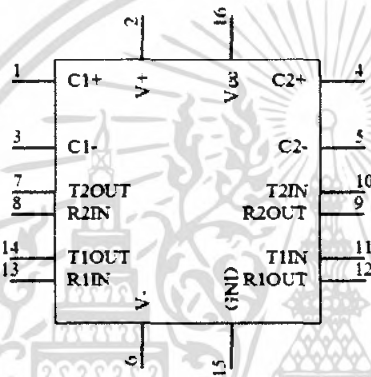
บทที่ 3

วงจรที่ใช้งานและการออกแบบโปรแกรม

3.1 วงจรที่ใช้งาน

วงจรที่ใช้ส่งข้อมูลจากคอมพิวเตอร์ไปยัง AT89C51

3.1.1 ไอซี MAX232 เราจะใช้ไอซี MAX232 เป็นตัวรับข้อมูลจากคอมพิวเตอร์แล้วส่งไปยังไมโครคอนโทรลเลอร์ โดยไอซี MAX232 มีความเร็วในการส่งข้อมูลเท่ากับ 120 กิโลบิต ต่อ 1 วินาที (kbps)



รูป 3.1 แสดงขาของไอซี MAX232

ตารางที่ 3.1 ตารางแสดงหน้าที่การทำงานของขาต่างๆ ของ MAX232

ขา	ชื่อ	รายละเอียด
1,3	C1+,C1-	Terminals for positive charge-pump capacitor
2	V+	+2VCC voltage generated by the charge pump
4,5	C2+,C2-	Terminals for negative charge-pump capacitor
6	V-	-2VCC voltage generated by the charge pump
7,14	T2OUT	RS-232 Driver Outputs
8,13	T1OUT	RS-232 Receiver Inputs
9,12	R2OUT,R1OUT	RS-232 Receiver Outputs
10,11	T2IN,T1IN	RS-232 Driver Inputs
15	GND	Ground
16	VCC	+4.5V to +5.5V Supply – Voltage Input

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.2 ไมโครคอนโทรลเลอร์ เราจะใช้เป็น AT89C51 ซึ่งมีพอร์ตที่ใช้งาน 2 พอร์ตคือพอร์ต 1 และ พอร์ต 3 โดยที่พอร์ต 3 นั้นจะเป็นอินพุท/เอาต์พุทพอร์ต (I/O Port) แล้วยังมีคุณสมบัติพิเศษดังนี้

1	P1.0	Vcc	40
2	P1.1	(AD0) P0.0	39
3	P1.2	(AD1) P0.1	38
4	P1.3	(AD2) P0.2	37
5	P1.4	(AD3) P0.3	36
6	P1.5	(AD4) P0.4	35
7	P1.6	(AD5) P0.5	34
8	P1.7	(AD6) P0.6	33
9	RST	(AD7) P0.7	32
10	P3.0 (RXD)	(Vpp) EA	31
11	P3.1 (TXD)	(PROG) ALE	30
12	P3.2 ($\overline{\text{INT0}}$)	PSEN	29
13	P3.3 ($\overline{\text{INT1}}$)	(A15) P2.7	28
14	P3.4 (T0)	(A14) P2.6	27
15	P3.5 (T1)	(A13) P2.5	26
16	P3.6 ($\overline{\text{WR}}$)	(A12) P2.4	25
17	P3.7 ($\overline{\text{RD}}$)	(A11) P2.3	24
18	XTAL2	(A10) P2.2	23
19	XTAL1	(A9) P2.1	22
20	GND	(A8) P2.0	21

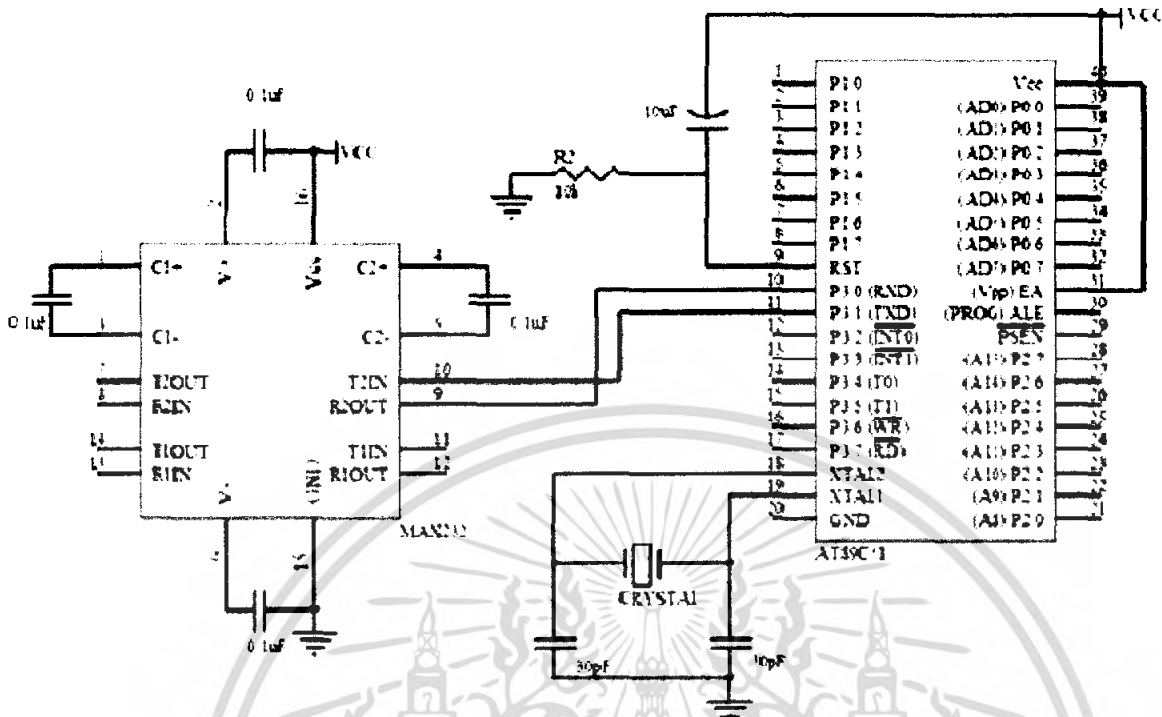
AT89C51

รูป 3.2 แสดงขาไอซี AT89C51

ตารางที่ 3.2 ตารางแสดงคุณสมบัติพิเศษของพอร์ต 3 ของ AT89C51

Port Pin	Alternate Function
P3.0	RXD (serial input port)
P3.1	TXD (Serial output port)
P3.2	$\overline{\text{INT0}}$ (external interrupt 0)
P3.3	$\overline{\text{INT1}}$ (external interrupt 1)
P3.4	T0(timer 0 external input)
P3.5	T1(timer 1 external input)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

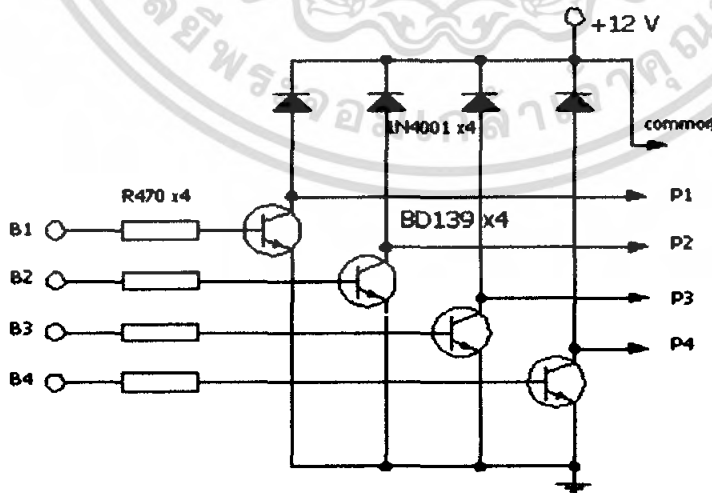


รูป 3.3 วงจรที่ใช้ส่งข้อมูลจากคอมพิวเตอร์ไปยังไมโครคอนโทรลเลอร์

จากรูปที่ 3.3 ไอซี MAX232 จะเป็นตัวรับข้อมูลผ่านทางพอร์ตอนุกรมเข้ามาทางขา T2OUT และ R2IN และส่งข้อมูลไปยังไมโครคอนโทรลเลอร์ AT89C51 ทาง T2IN และ R2OUT

3.1.3 วงจรขับเคลื่อนมอเตอร์

วงจรขับเคลื่อนมอเตอร์ในโครงงานนี้ใช้เป็นวงจรทรานซิสเตอร์ โดยใช้ BD139 ทำหน้าที่ในการขับเคลื่อนมอเตอร์แต่ละเฟส



รูป 3.4 วงจรที่ใช้ในการขับเคลื่อนมอเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

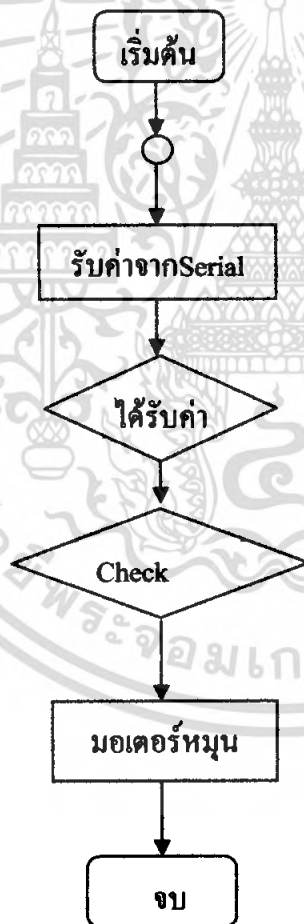
3.2 การออกแบบโปรแกรม

ในส่วนของโปรแกรมนั้น จะมาสามารถแบ่งออกได้เป็น ส่วนใหญ่ๆ คือ โปรแกรมในการควบคุมการทำงานของมอเตอร์ โปรแกรมควบคุมการทำงานของกล้อง และ โปรแกรมในส่วนของการประมวลผลภาพ

3.2.1 โปรแกรมในการควบคุมการหมุนของมอเตอร์

สำหรับโปรแกรมที่ใช้ในการควบคุมการหมุนของมอเตอร์นั้นจะควบคุมผ่านพอร์ตอนุกรม โดยใช้โปรแกรม C++ บิลเดอร์ โดยที่เราต้องการให้มอเตอร์นั้นทำงานได้ 2 ลักษณะ คือ

1. หมุนในลักษณะเป็นสเต็ปแล้วหยุดตามขนาดองศาที่กำหนดเพื่อให้กล้องจับภาพเป็นภาพนิ่ง
2. หมุนในลักษณะต่อเนื่องจนครบ 1 รอบเพื่อให้กล้องจับภาพเป็นภาพวิดีโอ



รูป 3.5 แผนภาพการทำงานของมอเตอร์ที่รับค่าจากพอร์ตอนุกรม

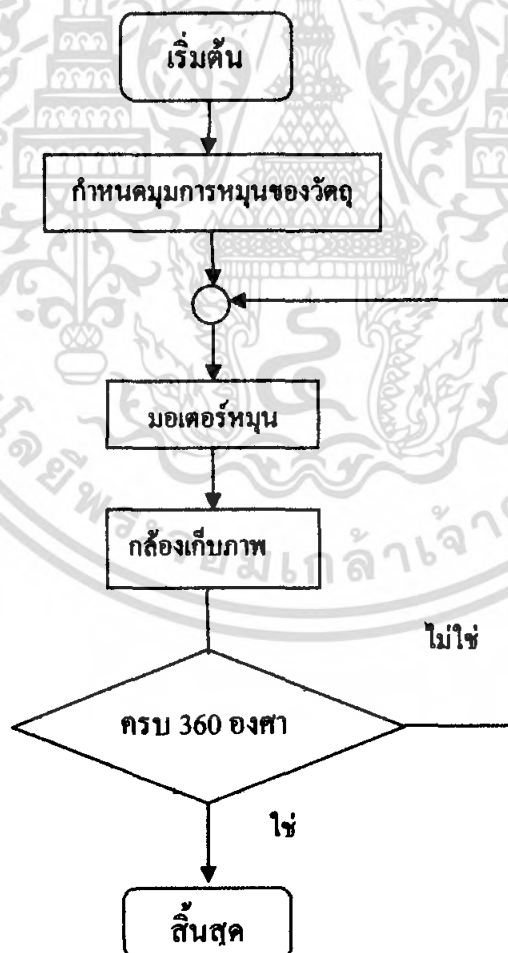
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.2 โปรแกรมการประมวลผลภาพ

จะใช้โปรแกรม C++Builder เป็นตัวประมวลผล การใช้งานนั้นสามารถแบ่งออกได้เป็น 3 ขั้นตอนได้แก่

1.การเก็บภาพ(Acquisition)

เป็นการจัดเก็บข้อมูลเบื้องต้นที่ต้องการจะนำไปประมวลผล โดยการแสดงผลภาพเบื้องต้นนั้น จะใช้โปรแกรม C++Builder โดยใช้ component VideoOCX ซึ่งเป็นซอฟต์แวร์ที่ใช้สำหรับการแสดงผลภาพโดยเฉพาะ โดยเราต้องการโปรแกรมที่สามารถแสดงภาพที่รับจากกล้องให้เห็นได้ชัดเจน สามารถที่จะรับภาพนั้นเป็นภาพนิ่งได้ แล้วนำภาพที่รับได้มาบันทึกเก็บไว้ใน Folder ที่กำหนด ในขณะที่ช่วงกันส่วนนี้จะต้องควบคุมการทำงานของมอเตอร์และกล้องไปในเวลาเดียวกัน เพื่อที่กล้องจะบันทึกภาพในขณะที่มอเตอร์หมุนที่มุมต่างๆ ทำให้ได้รูปวัตถุที่มุมต่างๆ โปรแกรมในส่วนการเก็บภาพมีบล็อกไดอะแกรมแสดงการทำงานดังนี้

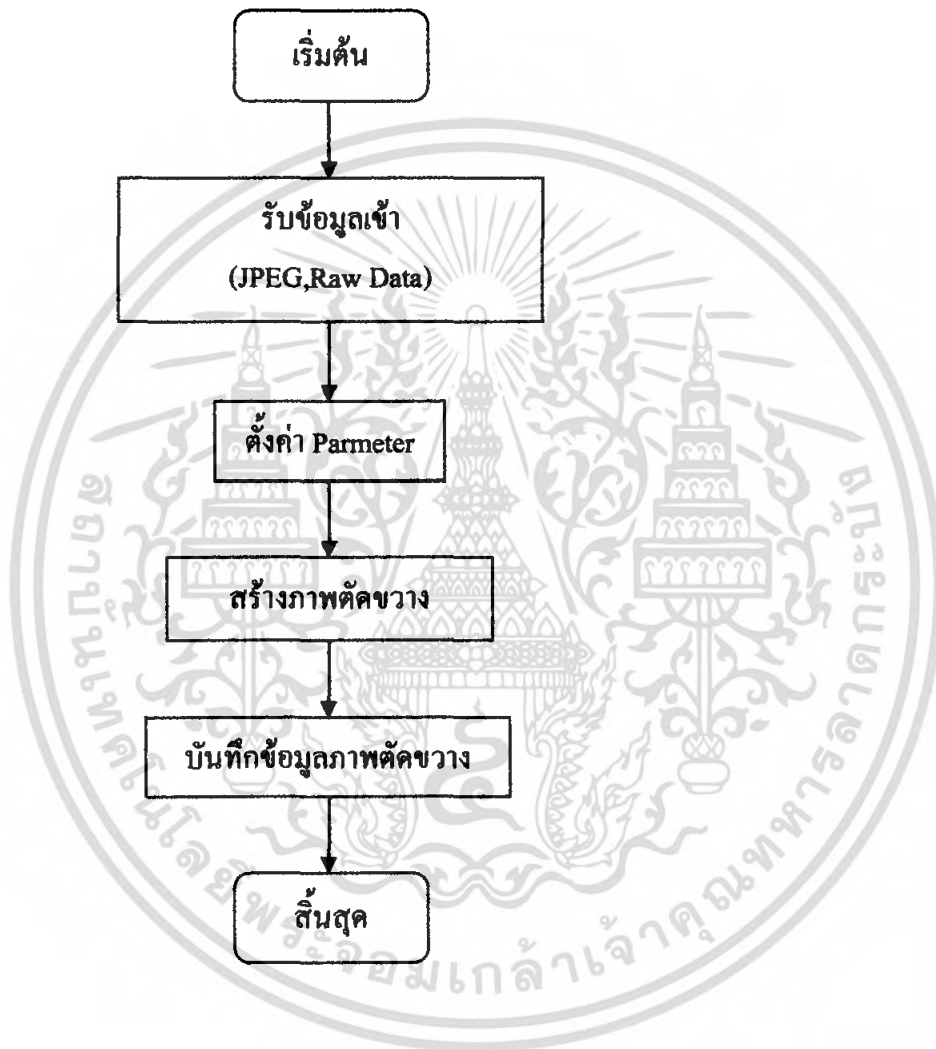


รูป 3.6 บล็อกไดอะแกรมแสดงการทำงานของโปรแกรมส่วนการเก็บภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.การสร้างภาพตัดขวาง(Reconstruction)

เป็นกระบวนการทำงานที่จะรับข้อมูลจากในขั้นตอนแรกนั่นคือหลังจากเก็บภาพที่ต้องการจะประมวลผลเรียบร้อยแล้ว นำเข้าสู่โปรแกรมในส่วนการสร้างภาพตัดขวาง เพื่อการประมวลผลออกมา สิ่งที่ได้จากขั้นตอนนี้คือ ภาพตัดขวางที่สามารถนำไปสร้างแบบจำลอง 3 มิติได้

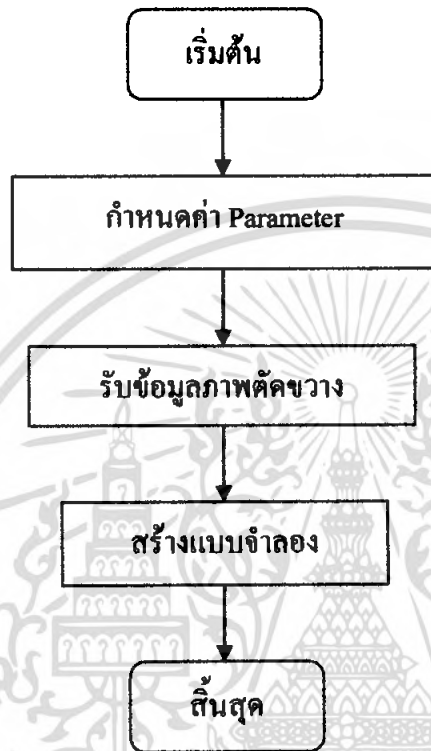


รูป 3.7 บล็อก ไคอะแกรมแสดงการทำงานของโปรแกรมส่วนการสร้างภาพตัดขวาง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.การสร้างแบบจำลอง 3 มิติ(3D Modelling)

เป็นกระบวนการสุดท้ายโดยรับข้อมูลเป็นภาพตัดขวางของวัตถุ แล้วนำมาสร้างเป็นแบบจำลอง 3 มิติ



รูป 3.8 ไคอะแกรมแสดงการทำงานของโปรแกรมส่วนการสร้างแบบจำลอง 3 มิติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การทดลอง

4.1 การทดลองส่วนของการควบคุมสเต็ปมอเตอร์

จากการทดลองการควบคุมสเต็ปมอเตอร์ พบว่าไมโครคอนโทรลเลอร์สามารถรับข้อมูลผ่านทางพอร์ตอนุกรมจากคอมพิวเตอร์ได้ และสามารถบังคับการหมุนของมอเตอร์ในแบบสเต็ปได้ค่าองศาต่างๆ

ตารางที่ 4.1 ขนาดมุมของมอเตอร์และจำนวนภาพที่ได้

มุม	จำนวนภาพ	มุม	จำนวนภาพ
0.9	401	36.0	11
1.8	201	45.0	9
3.6	101	72.0	6
4.5	81	90.0	5
9.0	41	180.0	3
18.0	21	360.0	2

ตารางที่ 4.2 แสดงค่าของมุมที่กำหนดและมุมที่ได้จริง

ค่าที่กำหนด	ค่าที่ได้จริง	ค่าที่กำหนด	ค่าที่ได้จริง
0.9	1.0	36.0	36.5
1.8	1.8	45.0	45.0
3.6	3.5	72.0	71.5
4.5	4.5	90.0	90.5
9.0	9.0	180.0	180.0
18.0	18.5	360.0	359.5

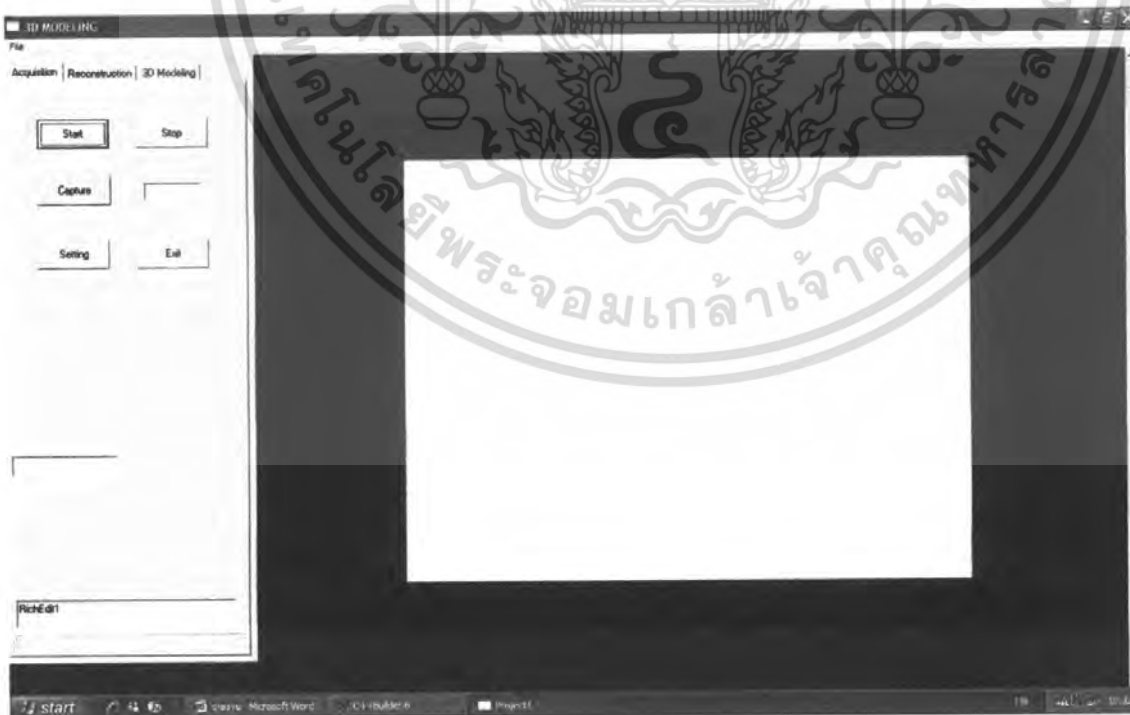
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.3 ตารางแสดงความแม่นยำของมุมเมื่อมีการทำซ้ำ

มุม	ครั้งที่ 1	ครั้งที่ 2	ครั้งที่ 3	ครั้งที่ 4	ครั้งที่ 5	ครั้งที่ 6	ครั้งที่ 7	ครั้งที่ 8	ครั้งที่ 9	ครั้งที่ 10
45.0	45.0	45.5	45.0	45.5	45.5	44.5	45.5	46.5	44.5	45.0
90.0	90.5	89.5	89.0	90.0	90.5	90.5	89.5	90.0	91.0	90.5
135.0	134.5	135.0	135.0	134.0	135.5	134.5	135.5	136.5	134.0	135.0
180.0	179.5	180.0	179.5	180.0	179.5	179.5	180.0	179.5	180.5	179.5
225.0	225.0	225.5	225.0	224.5	225.0	225.0	224.5	225.5	224.5	225.0
270.0	270.0	270.0	271.0	270.0	270.5	269.5	270.0	270.5	269.5	269.5
315.0	314.5	314.5	315.0	315.5	315.0	315.0	315.5	314.0	314.5	315.0
360.0	360.0	359.5	360.5	360.0	361.0	359.5	359.5	360.0	360.0	360.5

4.2 การทดลองส่วนการรับภาพและบันทึกภาพ

โปรแกรมที่ใช้สำหรับรับภาพและควบคุมการหมุนของมอเตอร์เป็นดังรูปที่แสดงนี้ มีปุ่มในการทำงานต่างดังนี้คือ



รูป 4.1 โปรแกรมที่ออกแบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Start โปรแกรมจะเริ่มจับภาพจากกล้องแสดงออกมาบนจอ

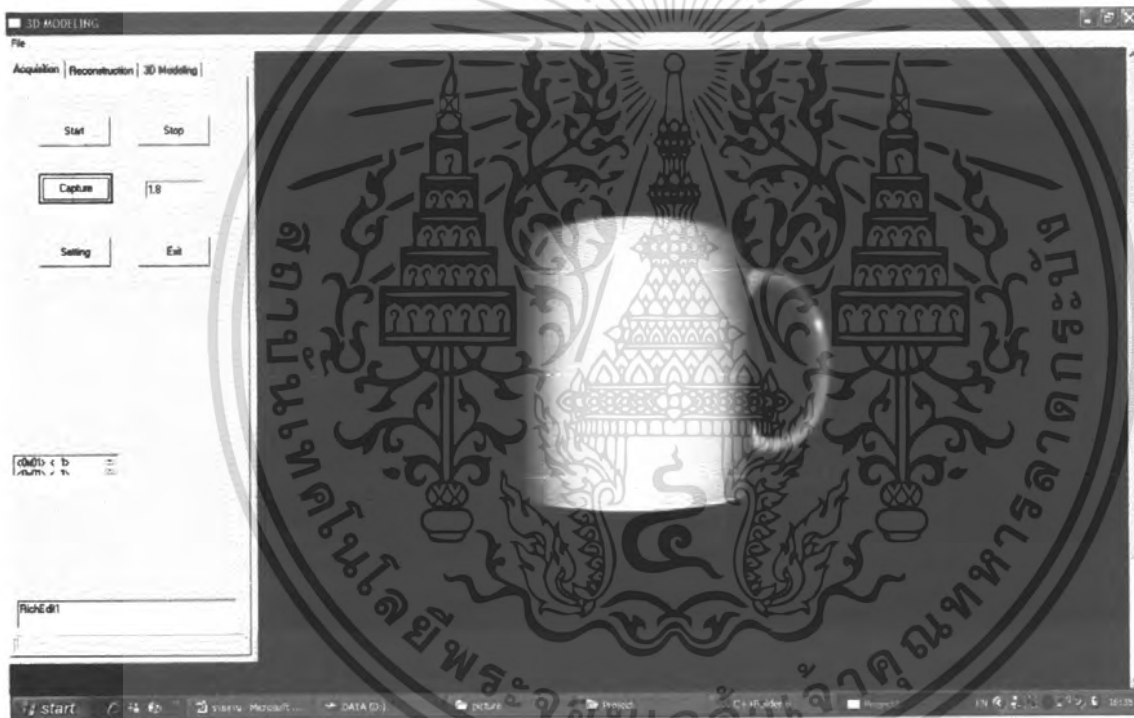
Image เป็นปุ่มที่กำหนดการทำงานของ โปรแกรมให้มอเตอร์หมุนและกล้องเริ่มบันทึกภาพ

Setting ใช้สำหรับการปรับแต่งค่าต่างๆ ของกล้อง เช่น ความคมชัด ภาพสี ขาวดำ ความสว่าง เป็นต้น

Stop ใช้ในการหยุดการทำงานของกล้องในการจับภาพ

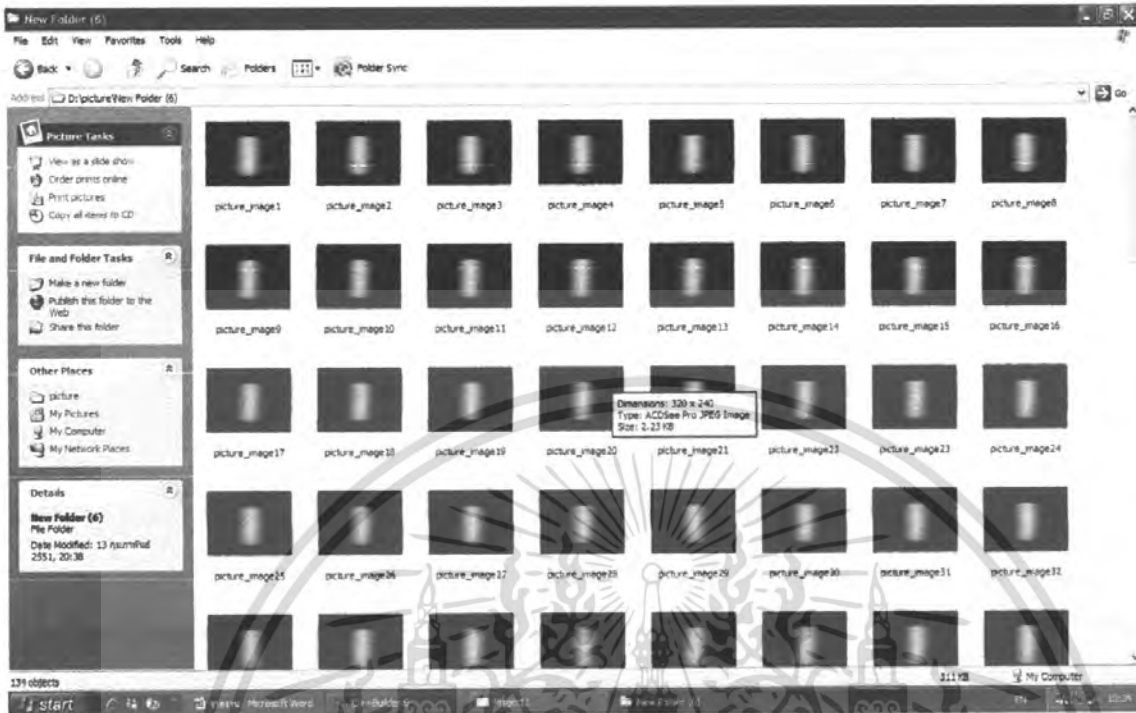
Exit ใช้ในการออกจากโปรแกรม

สำหรับการทดลองในส่วนของการรับภาพของกล้องนั้น สามารถที่จะรับภาพของวัตถุได้และนำภาพที่ได้นั้นมาจัดเก็บไว้ใน Folder ที่กำหนดได้



รูปที่ 4.2 แสดงการรับภาพจากกล้องแสดงบน โปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.3 แสดงข้อมูลภาพที่ได้และบันทึกใน Folder ที่กำหนด

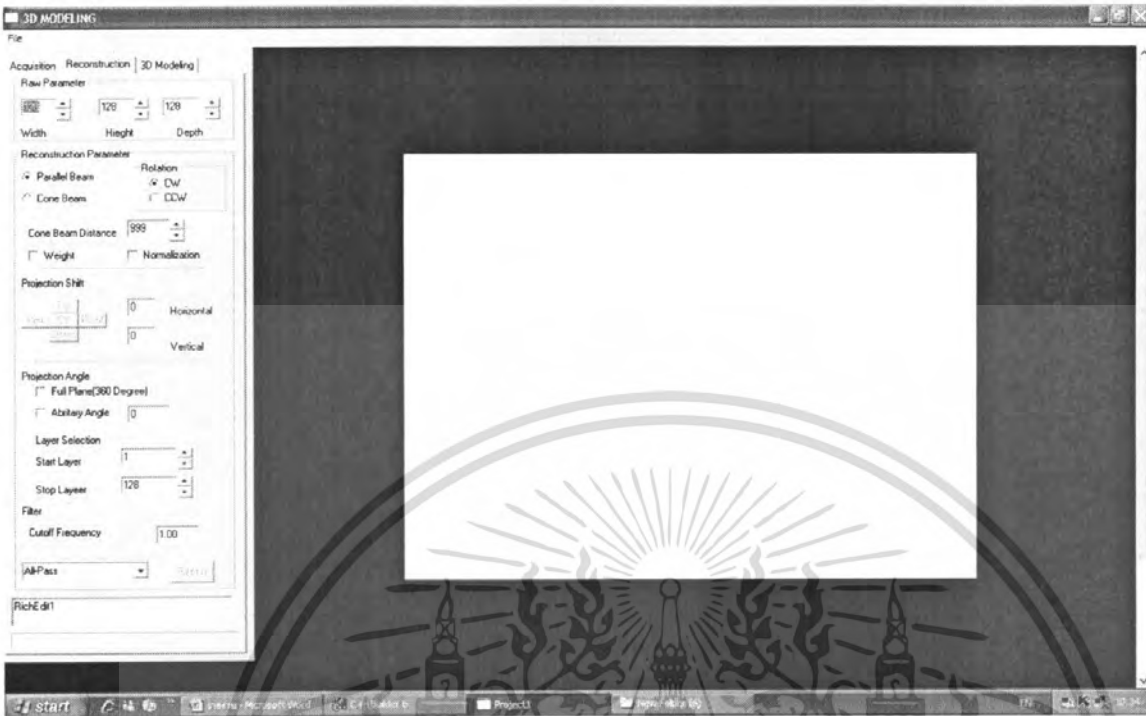
เมื่อกดปุ่ม Setting จะปรากฏ Dialog box ในการปรับค่าต่างๆ ของกล้อง



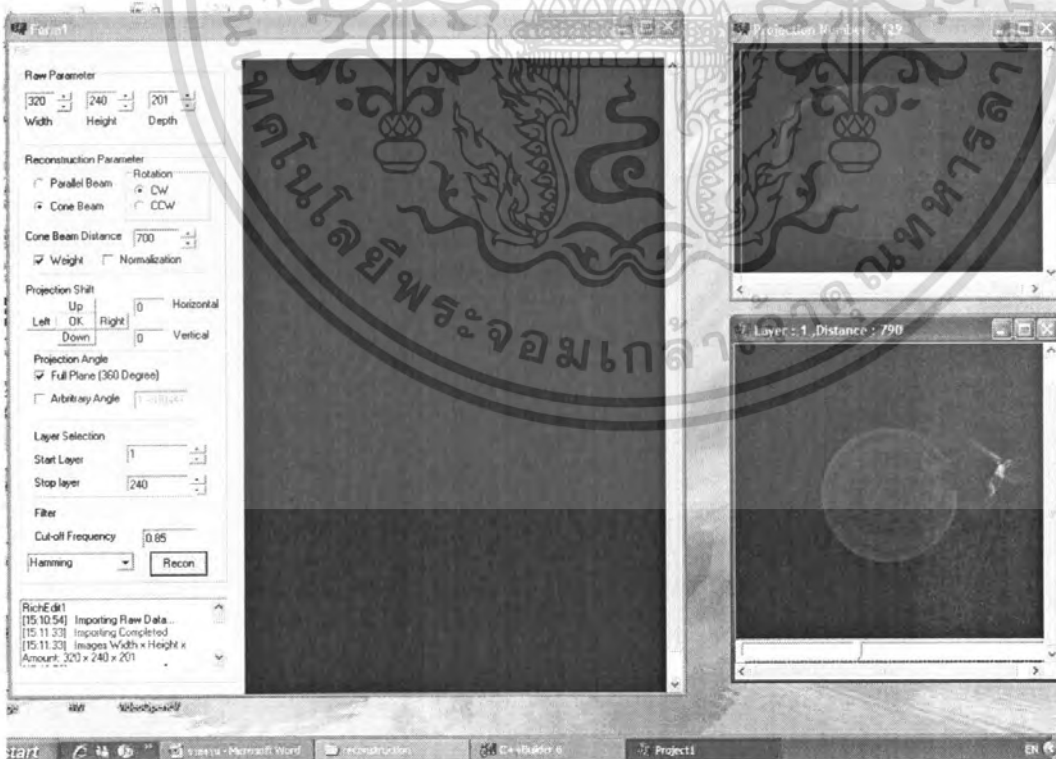
รูปที่ 4.4 Dialog box ปรับแต่งค่าของกล้อง

ในส่วนของการจับภาพนี้เราจะบันทึกภาพที่ได้เก็บไว้แล้วนำไปประมวลผลภาพต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

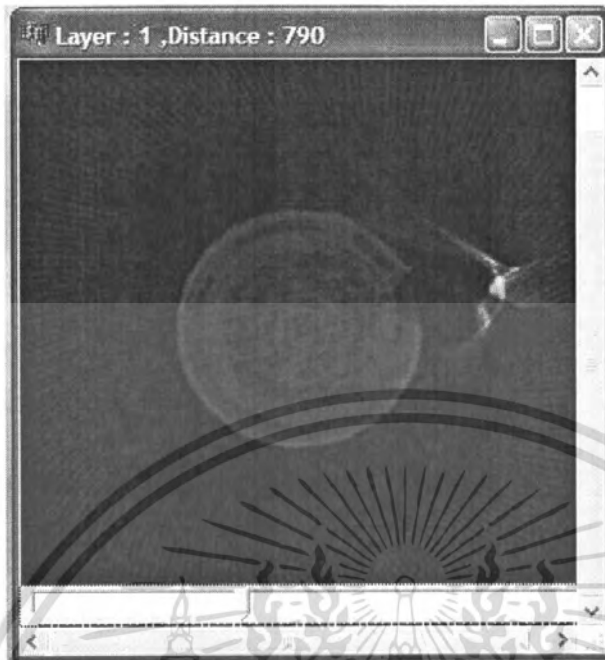


รูปที่ 4.5 แสดงโปรแกรมส่วนของการสร้างภาพตัดขวาง

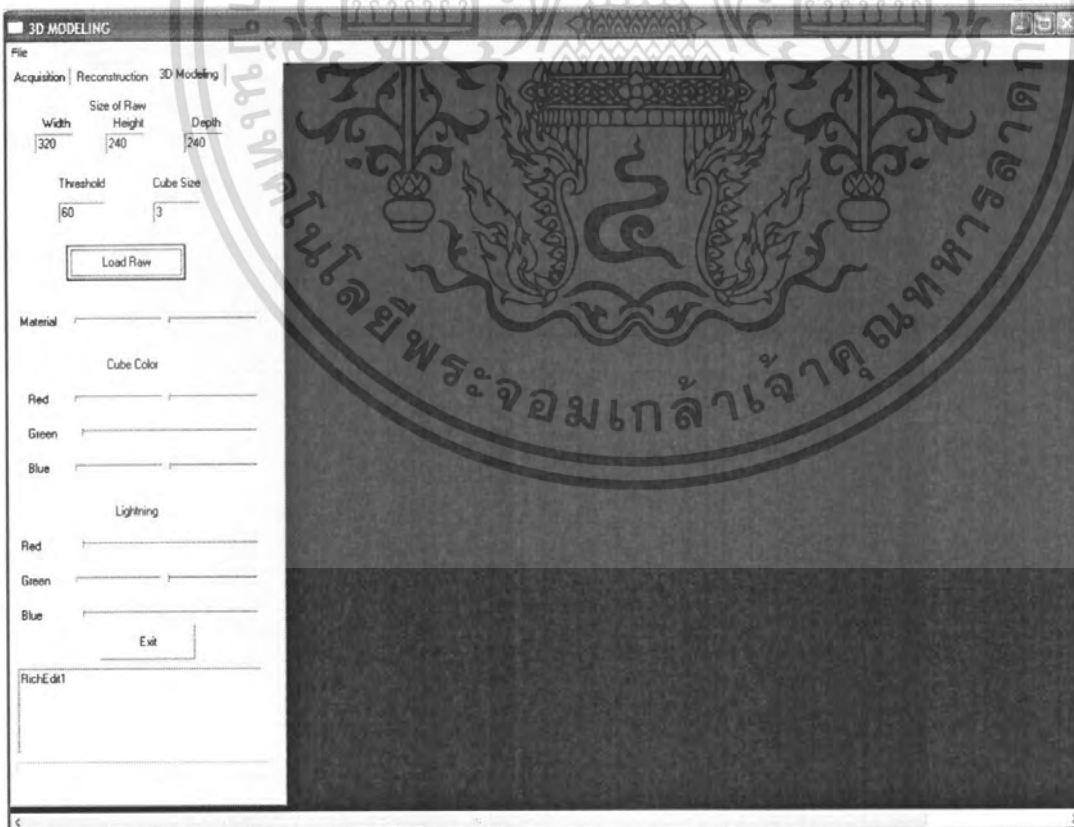


รูปที่ 4.6 แสดงการใช้งานโปรแกรมในส่วนของการสร้างภาพตัดขวาง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

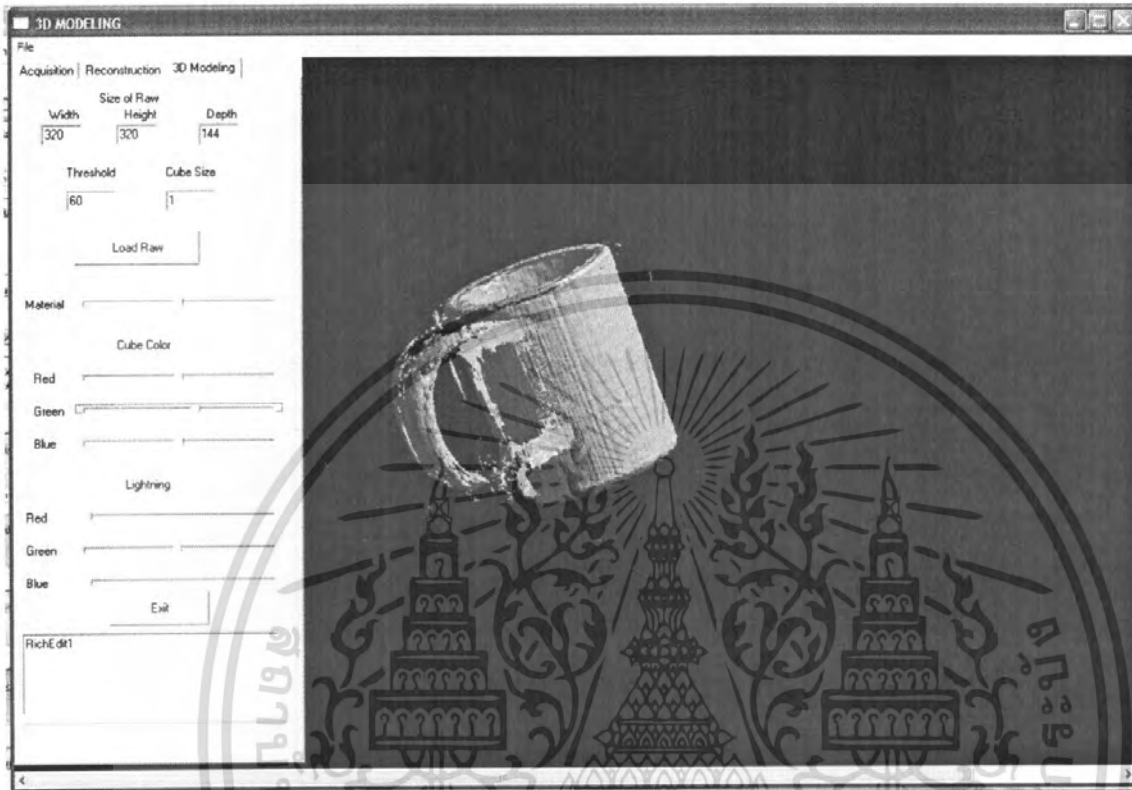


รูปที่ 4.7 แสดงภาพตัดขวางที่ได้จากการใช้โปรแกรม



รูปที่ 4.8 แสดงโปรแกรมส่วนของการสร้างแบบจำลอง 3 มิติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.9 แสดงแบบจำลอง 3 มิติที่ได้จากโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

สรุปและวิเคราะห์ผลการดำเนินงาน

5.1 สรุปและวิเคราะห์ผลการทดลอง

จากการทดลองพบว่า

1. มอเตอร์สามารถหมุนได้ตามขนาดมุมที่กำหนดและกล้องสามารถจัดเก็บภาพวัตถุที่มุมต่างๆ และบันทึกไว้ใน Folder ที่ต้องการได้
2. ภาพที่บันทึกได้จากกล้องสามารถนำมาสร้างเป็นภาพตัดขวางโดยการใช้โปรแกรมสร้างภาพตัดขวางได้
3. สามารถสร้างแบบจำลอง 3 มิติจากภาพตัดขวางโดยโปรแกรม 3D Modelling ได้

5.2 ปัญหาในการดำเนินงาน

1. เนื่องจากกล้องมีความละเอียดไม่สูงมากนัก ทำให้ภาพที่ได้รับคุณภาพความคมชัดไม่สูงเช่นกัน
2. ภาพตัดขวางที่ได้ไม่สมบูรณ์ เนื่องจากปัญหาด้านสภาพแวดล้อมขณะบันทึกภาพ คือ แสงจากปัจจัยภายนอก ดังนั้นเมื่อนำมาจำลองเป็นภาพ 3 มิติ จึงได้ภาพที่ไม่สมบูรณ์

5.3 แนวทางในการพัฒนาต่อ

ปรับปรุงเทคนิคที่ใช้ในการรับภาพ เพื่อให้ภาพที่ได้รับมีความคมชัดมากขึ้น เพื่อส่งผลการนำภาพที่ได้รับไปประมวลผลต่อไป และพัฒนาโปรแกรมให้มีการใช้งานที่สะดวกมากขึ้น

หนังสืออ้างอิง

- [1] นคร ภักดีชาติ, วีรบุลย์ หล่อวิเชียรรุ่ง, ชัยวัฒน์ ถัมพรจิตรวิไล, “ปฏิบัติการไมโครคอนโทรลเลอร์ MCS-51 ด้วยโปรแกรมภาษาซี”
- [2] นฤต กระจาย “เริ่มต้นใช้งาน C++Builder”
- [3] รศ.ดร.ชูชาติ ปิณฑวิรุจน์ “เอกสารประกอบการสอนวิชา Digital Image Processing”



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

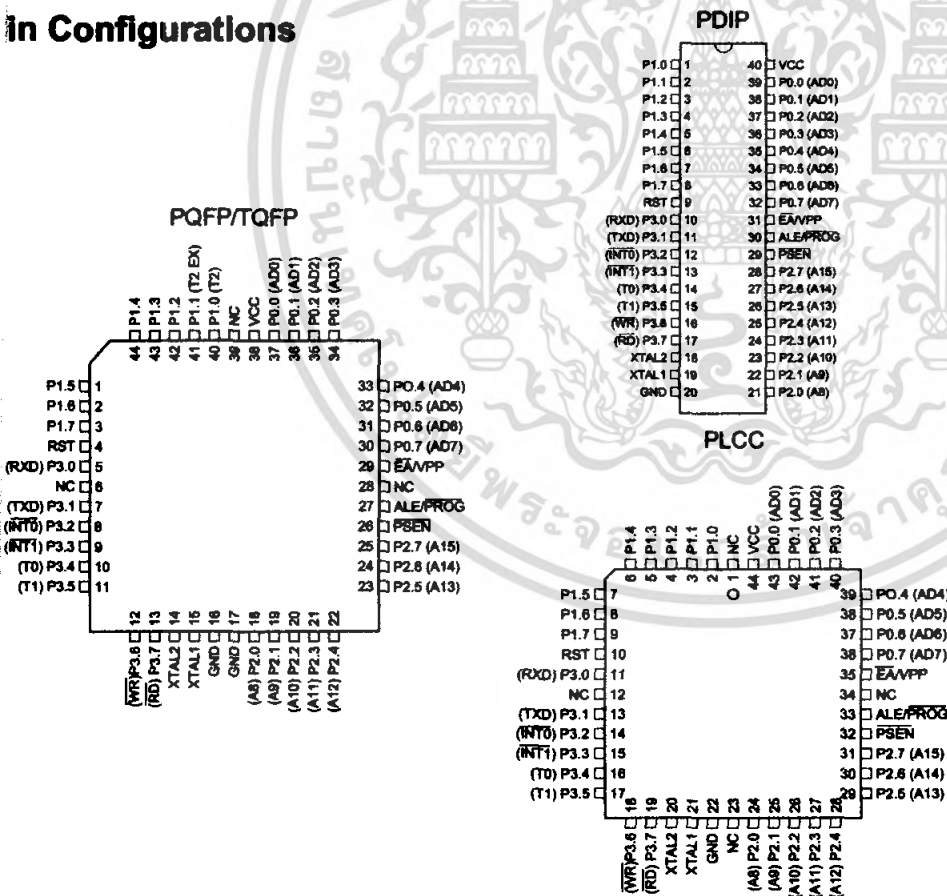
Features

- Compatible with MCS-51™ Products
- 4K Bytes of In-System Reprogrammable Flash Memory
 - Endurance: 1,000 Write/Erase Cycles
- Fully Static Operation: 0 Hz to 24 MHz
- Three-level Program Memory Lock
- 128 x 8-bit Internal RAM
- 32 Programmable I/O Lines
- Two 16-bit Timer/Counters
- Six Interrupt Sources
- Programmable Serial Channel
- Low-power Idle and Power-down Modes

Description

The AT89C51 is a low-power, high-performance CMOS 8-bit microcomputer with 4K bytes of Flash programmable and erasable read only memory (PEROM). The device is manufactured using Atmel's high-density nonvolatile memory technology and is compatible with the industry-standard MCS-51 instruction set and pinout. The on-chip Flash allows the program memory to be reprogrammed in-system or by a conventional nonvolatile memory programmer. By combining a versatile 8-bit CPU with Flash on a monolithic chip, the Atmel AT89C51 is a powerful microcomputer which provides a highly-flexible and cost-effective solution to many embedded control applications.

Pin Configurations



8-bit Microcontroller with 4K Bytes Flash

AT89C51

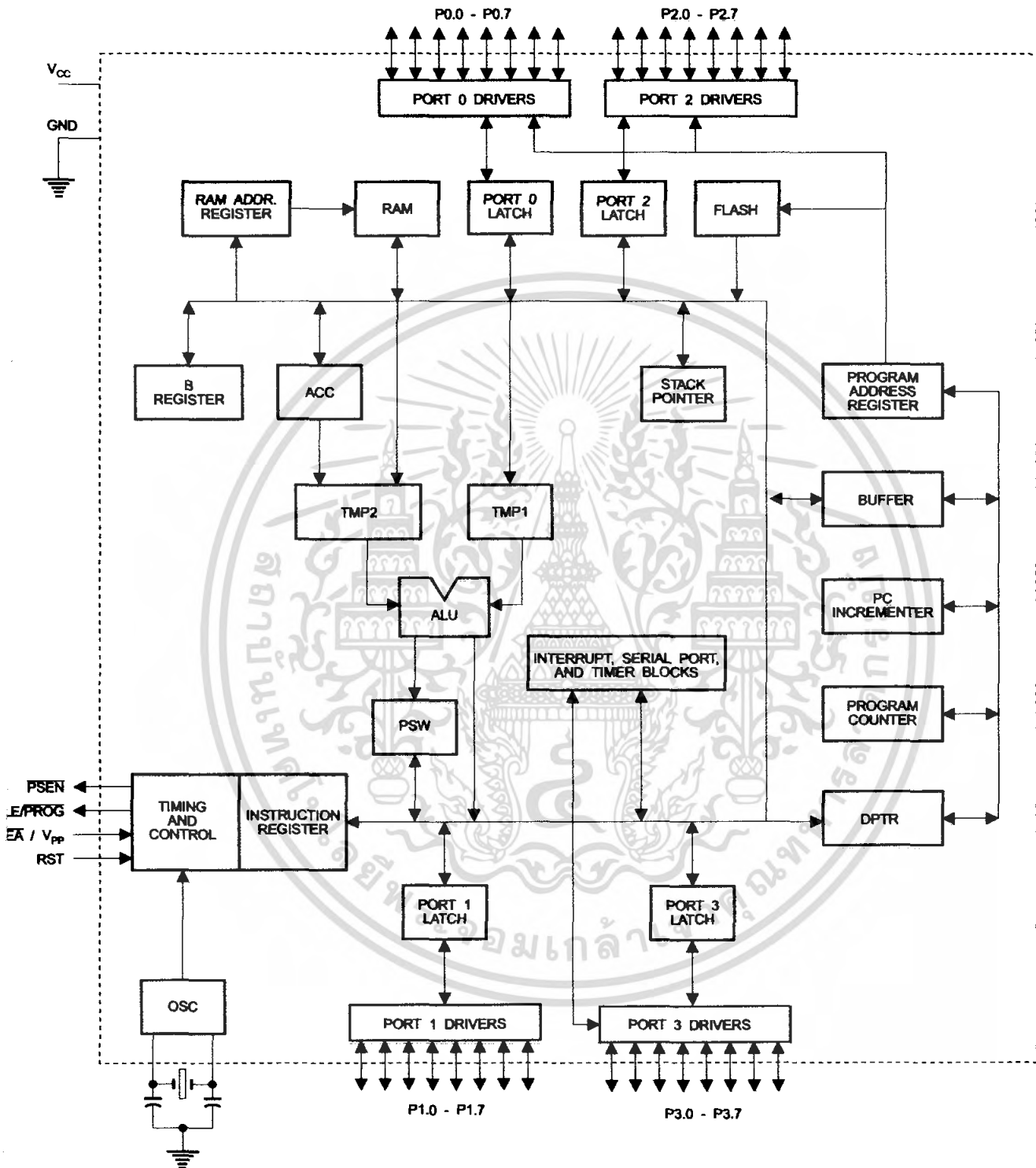
**Not Recommended
for New Designs.
Use AT89S51.**

Rev. 0285G-02/00



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Block Diagram



AT89C51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

The AT89C51 provides the following standard features: 4K bytes of Flash, 128 bytes of RAM, 32 I/O lines, two 16-bit timer/counters, a five vector two-level interrupt architecture, full duplex serial port, on-chip oscillator and clock circuitry. In addition, the AT89C51 is designed with static logic operation down to zero frequency and supports two software selectable power saving modes. The Idle Mode stops the CPU while allowing the RAM, timer/counters, serial port and interrupt system to continue functioning. The Power-down Mode saves the RAM contents but freezes the oscillator disabling all other chip functions until the next hardware reset.

Port 2 pins that are externally being pulled low will source current (I_{IL}) because of the internal pullups.

Port 2 emits the high-order address byte during fetches from external program memory and during accesses to external data memory that use 16-bit addresses (MOVX @ DPTR). In this application, it uses strong internal pullups when emitting 1s. During accesses to external data memory that use 8-bit addresses (MOVX @ RI), Port 2 emits the contents of the P2 Special Function Register.

Port 2 also receives the high-order address bits and some control signals during Flash programming and verification.

Port 3

Port 3 is an 8-bit bi-directional I/O port with internal pullups. The Port 3 output buffers can sink/source four TTL inputs. When 1s are written to Port 3 pins they are pulled high by the internal pullups and can be used as inputs. As inputs, Port 3 pins that are externally being pulled low will source current (I_{IL}) because of the pullups.

Port 3 also serves the functions of various special features of the AT89C51 as listed below:

Port Pin	Alternate Functions
P3.0	RXD (serial input port)
P3.1	TXD (serial output port)
P3.2	$\overline{INT0}$ (external interrupt 0)
P3.3	$\overline{INT1}$ (external interrupt 1)
P3.4	T0 (timer 0 external input)
P3.5	T1 (timer 1 external input)
P3.6	\overline{WR} (external data memory write strobe)
P3.7	\overline{RD} (external data memory read strobe)

Port 3 also receives some control signals for Flash programming and verification.

RST

Reset input. A high on this pin for two machine cycles while the oscillator is running resets the device.

ALE/PROG

Address Latch Enable output pulse for latching the low byte of the address during accesses to external memory. This pin is also the program pulse input (\overline{PROG}) during Flash programming.

In normal operation ALE is emitted at a constant rate of 1/6 the oscillator frequency, and may be used for external timing or clocking purposes. Note, however, that one ALE

In Description

DC

Supply voltage.

ND

Ground.

Port 0

Port 0 is an 8-bit open-drain bi-directional I/O port. As an output port, each pin can sink eight TTL inputs. When 1s are written to port 0 pins, the pins can be used as high-impedance inputs.

Port 0 may also be configured to be the multiplexed low-order address/data bus during accesses to external program and data memory. In this mode P0 has internal pullups.

Port 0 also receives the code bytes during Flash programming, and outputs the code bytes during program verification. External pullups are required during program verification.

Port 1

Port 1 is an 8-bit bi-directional I/O port with internal pullups. The Port 1 output buffers can sink/source four TTL inputs. When 1s are written to Port 1 pins they are pulled high by the internal pullups and can be used as inputs. As inputs, Port 1 pins that are externally being pulled low will source current (I_{IL}) because of the internal pullups.

Port 1 also receives the low-order address bytes during Flash programming and verification.

Port 2

Port 2 is an 8-bit bi-directional I/O port with internal pullups. The Port 2 output buffers can sink/source four TTL inputs. When 1s are written to Port 2 pins they are pulled high by the internal pullups and can be used as inputs. As inputs,



Use is skipped during each access to external Data Memory.

As desired, ALE operation can be disabled by setting bit 0 of PFR location 8EH. With the bit set, ALE is active only during a MOVX or MOVC instruction. Otherwise, the pin is weakly pulled high. Setting the ALE-disable bit has no effect if the microcontroller is in external execution mode.

PSEN

Program Store Enable is the read strobe to external program memory.

When the AT89C51 is executing code from external program memory, PSEN is activated twice each machine cycle, except that two PSEN activations are skipped during each access to external data memory.

EA/VPP

External Access Enable. EA must be strapped to GND in order to enable the device to fetch code from external program memory locations starting at 0000H up to FFFFH. Note, however, that if lock bit 1 is programmed, EA will be permanently latched on reset.

EA should be strapped to V_{CC} for internal program executions.

This pin also receives the 12-volt programming enable voltage (V_{PP}) during Flash programming, for parts that require 2-volt V_{PP}.

XTAL1

Input to the inverting oscillator amplifier and input to the external clock operating circuit.

XTAL2

Output from the inverting oscillator amplifier.

Oscillator Characteristics

XTAL1 and XTAL2 are the input and output, respectively, of an inverting amplifier which can be configured for use as an on-chip oscillator, as shown in Figure 1. Either a quartz crystal or ceramic resonator may be used. To drive the device from an external clock source, XTAL2 should be left

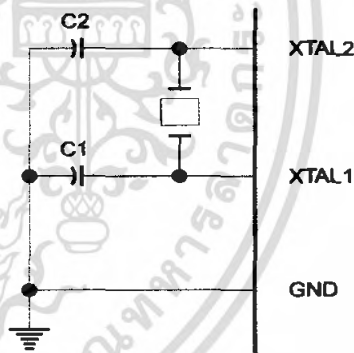
unconnected while XTAL1 is driven as shown in Figure 2. There are no requirements on the duty cycle of the external clock signal, since the input to the internal clocking circuitry is through a divide-by-two flip-flop, but minimum and maximum voltage high and low time specifications must be observed.

Idle Mode

In idle mode, the CPU puts itself to sleep while all the on-chip peripherals remain active. The mode is invoked by software. The content of the on-chip RAM and all the special functions registers remain unchanged during this mode. The idle mode can be terminated by any enabled interrupt or by a hardware reset.

It should be noted that when idle is terminated by a hardware reset, the device normally resumes program execution, from where it left off, up to two machine cycles before the internal reset algorithm takes control. On-chip hardware inhibits access to internal RAM in this event, but access to the port pins is not inhibited. To eliminate the possibility of an unexpected write to a port pin when Idle is terminated by reset, the instruction following the one that invokes Idle should not be one that writes to a port pin or to external memory.

Figure 1. Oscillator Connections



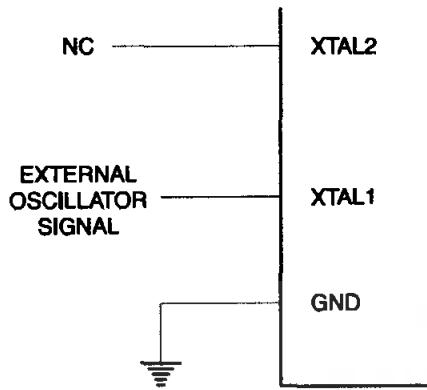
Note: C1, C2 = 30 pF ± 10 pF for Crystals
= 40 pF ± 10 pF for Ceramic Resonators

Status of External Pins During Idle and Power-down Modes

Mode	Program Memory	ALE	PSEN	PORT0	PORT1	PORT2	PORT3
Idle	Internal	1	1	Data	Data	Data	Data
Idle	External	1	1	Float	Data	Address	Data
Power-down	Internal	0	0	Data	Data	Data	Data
Power-down	External	0	0	Float	Data	Data	Data

AT89C51

Figure 2. External Clock Drive Configuration



ters retain their values until the power-down mode is terminated. The only exit from power-down is a hardware reset. Reset redefines the SFRs but does not change the on-chip RAM. The reset should not be activated before V_{CC} is restored to its normal operating level and must be held active long enough to allow the oscillator to restart and stabilize.

Program Memory Lock Bits

On the chip are three lock bits which can be left unprogrammed (U) or can be programmed (P) to obtain the additional features listed in the table below.

When lock bit 1 is programmed, the logic level at the \overline{EA} pin is sampled and latched during reset. If the device is powered up without a reset, the latch initializes to a random value, and holds that value until reset is activated. It is necessary that the latched value of \overline{EA} be in agreement with the current logic level at that pin in order for the device to function properly.

Power-down Mode

In the power-down mode, the oscillator is stopped, and the instruction that invokes power-down is the last instruction executed. The on-chip RAM and Special Function Registers

Lock Bit Protection Modes

	Program Lock Bits			Protection Type
	LB1	LB2	LB3	
1	U	U	U	No program lock features
2	P	U	U	MOVX instructions executed from external program memory are disabled from fetching code bytes from internal memory, \overline{EA} is sampled and latched on reset, and further programming of the Flash is disabled
3	P	P	U	Same as mode 2, also verify is disabled
4	P	P	P	Same as mode 3, also external execution is disabled



Programming the Flash

The AT89C51 is normally shipped with the on-chip Flash memory array in the erased state (that is, contents = FFH) and ready to be programmed. The programming interface accepts either a high-voltage (12-volt) or a low-voltage (5V) program enable signal. The low-voltage programming mode provides a convenient way to program the AT89C51 inside the user's system, while the high-voltage programming mode is compatible with conventional third-party Flash or EPROM programmers.

The AT89C51 is shipped with either the high-voltage or low-voltage programming mode enabled. The respective top-side marking and device signature codes are listed in the following table.

	V _{PP} = 12V	V _{PP} = 5V
Top-side Mark	AT89C51 xxxx yyww	AT89C51 xxxx-5 yyww
Signature	(030H) = 1EH (031H) = 51H (032H) = FFH	(030H) = 1EH (031H) = 51H (032H) = 05H

The AT89C51 code memory array is programmed byte-by-byte in either programming mode. To program any non-bank byte in the on-chip Flash Memory, the entire memory must be erased using the Chip Erase Mode.

Programming Algorithm: Before programming the AT89C51, the address, data and control signals should be set up according to the Flash programming mode table and Figure 3 and Figure 4. To program the AT89C51, take the following steps.

1. Input the desired memory location on the address lines.
2. Input the appropriate data byte on the data lines.
3. Activate the correct combination of control signals.
4. Raise \overline{EA}/V_{PP} to 12V for the high-voltage programming mode.
5. Pulse ALE/PROG once to program a byte in the Flash array or the lock bits. The byte-write cycle is self-timed and typically takes no more than 1.5 ms. Repeat steps 1 through 5, changing the address

and data for the entire array or until the end of the object file is reached.

Data Polling: The AT89C51 features Data Polling to indicate the end of a write cycle. During a write cycle, an attempted read of the last byte written will result in the complement of the written datum on PO.7. Once the write cycle has been completed, true data are valid on all outputs, and the next cycle may begin. Data Polling may begin any time after a write cycle has been initiated.

Ready/Busy: The progress of byte programming can also be monitored by the RDY/BSY output signal. P3.4 is pulled low after ALE goes high during programming to indicate BUSY. P3.4 is pulled high again when programming is done to indicate READY.

Program Verify: If lock bits LB1 and LB2 have not been programmed, the programmed code data can be read back via the address and data lines for verification. The lock bits cannot be verified directly. Verification of the lock bits is achieved by observing that their features are enabled.

Chip Erase: The entire Flash array is erased electrically by using the proper combination of control signals and by holding ALE/PROG low for 10 ms. The code array is written with all "1"s. The chip erase operation must be executed before the code memory can be re-programmed.

Reading the Signature Bytes: The signature bytes are read by the same procedure as a normal verification of locations 030H, 031H, and 032H, except that P3.6 and P3.7 must be pulled to a logic low. The values returned are as follows.

- (030H) = 1EH indicates manufactured by Atmel
- (031H) = 51H indicates 89C51
- (032H) = FFH indicates 12V programming
- (032H) = 05H indicates 5V programming

Programming Interface

Every code byte in the Flash array can be written and the entire array can be erased by using the appropriate combination of control signals. The write operation cycle is self-timed and once initiated, will automatically time itself to completion.

All major programming vendors offer worldwide support for the Atmel microcontroller series. Please contact your local programming vendor for the appropriate software revision.

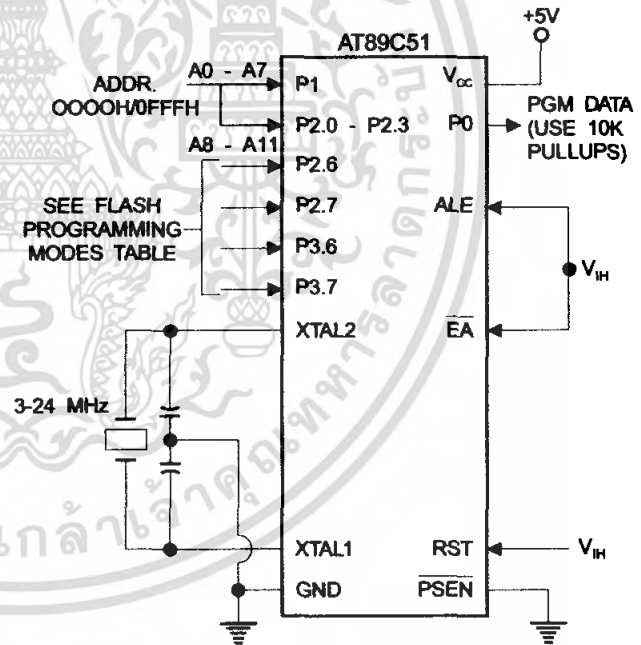
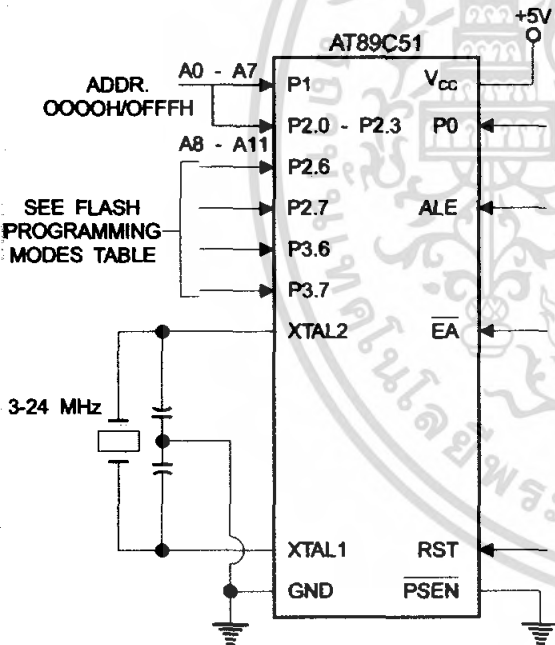
Flash Programming Modes

Mode	RST	PSEN	ALE/PROG	EA/V _{PP}	P2.6	P2.7	P3.6	P3.7
Write Code Data	H	L		H/12V	L	H	H	H
Read Code Data	H	L	H	H	L	L	H	H
Write Lock	H	L		H/12V	H	H	H	H
Chip Erase	H	L		H/12V	H	L	L	L
Read Signature Byte	H	L	H	H	L	L	L	L

Note: 1. Chip Erase requires a 10 ms PROG pulse.

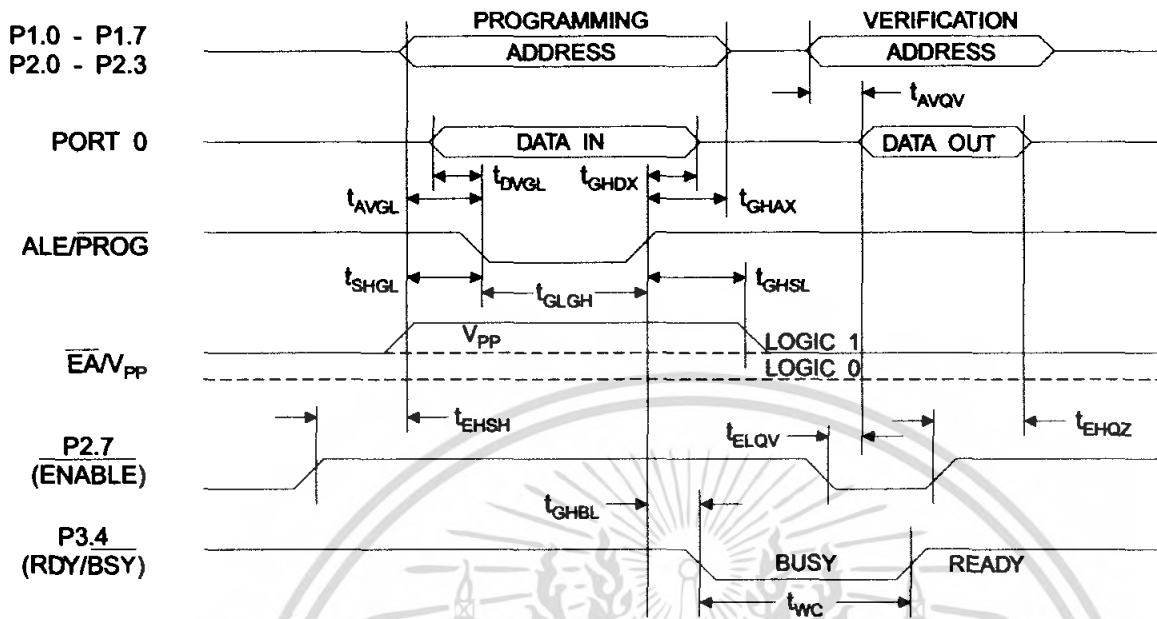
Figure 3. Programming the Flash

Figure 4. Verifying the Flash

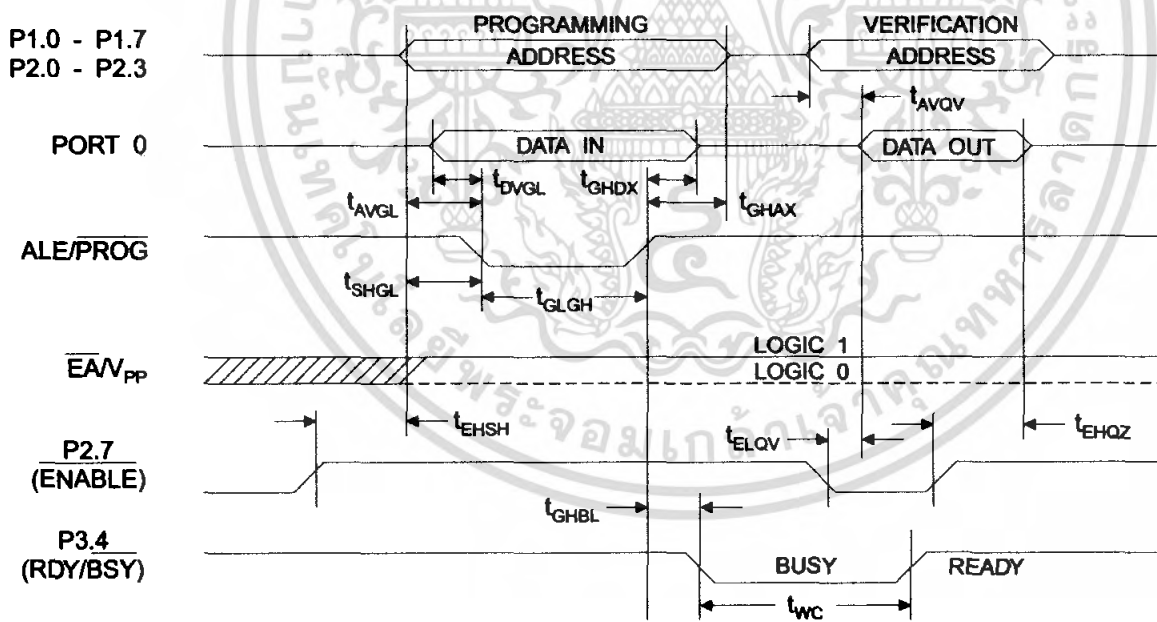




Flash Programming and Verification Waveforms - High-voltage Mode ($V_{PP} = 12V$)



Flash Programming and Verification Waveforms - Low-voltage Mode ($V_{PP} = 5V$)



Flash Programming and Verification Characteristics

= 0°C to 70°C, V_{CC} = 5.0 ± 10%

Symbol	Parameter	Min	Max	Units
V _{PP} ⁽¹⁾	Programming Enable Voltage	11.5	12.5	V
I _P ⁽¹⁾	Programming Enable Current		1.0	mA
f _{CLCL}	Oscillator Frequency	3	24	MHz
t _{WGL}	Address Setup to $\overline{\text{PROG}}$ Low	48t _{CLCL}		
t _{HAX}	Address Hold after $\overline{\text{PROG}}$	48t _{CLCL}		
t _{WGL}	Data Setup to $\overline{\text{PROG}}$ Low	48t _{CLCL}		
t _{HDX}	Data Hold after $\overline{\text{PROG}}$	48t _{CLCL}		
t _{ESH}	P2.7 ($\overline{\text{ENABLE}}$) High to V _{PP}	48t _{CLCL}		
t _{HGL}	V _{PP} Setup to $\overline{\text{PROG}}$ Low	10		μs
t _{HSL} ⁽¹⁾	V _{PP} Hold after $\overline{\text{PROG}}$	10		μs
t _{GLH}	$\overline{\text{PROG}}$ Width	1	110	μs
t _{AVQV}	Address to Data Valid		48t _{CLCL}	
t _{ELQV}	$\overline{\text{ENABLE}}$ Low to Data Valid		48t _{CLCL}	
t _{EQZ}	Data Float after $\overline{\text{ENABLE}}$	0	48t _{CLCL}	
t _{GHBL}	$\overline{\text{PROG}}$ High to $\overline{\text{BUSY}}$ Low		1.0	μs
t _{WC}	Byte Write Cycle Time		2.0	ms

Note: 1. Only used in 12-volt programming mode.



Absolute Maximum Ratings*

Operating Temperature.....	-55°C to +125°C
Storage Temperature.....	-65°C to +150°C
Voltage on Any Pin with Respect to Ground.....	-1.0V to +7.0V
Maximum Operating Voltage.....	6.6V
IO Output Current.....	15.0 mA

*NOTICE: Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions beyond those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

C Characteristics

= -40°C to 85°C, $V_{CC} = 5.0V \pm 20\%$ (unless otherwise noted)

Symbol	Parameter	Condition	Min	Max	Units
V_L	Input Low-voltage	(Except $\bar{E}A$)	-0.5	$0.2 V_{CC} - 0.1$	V
V_{L1}	Input Low-voltage ($\bar{E}A$)		-0.5	$0.2 V_{CC} - 0.3$	V
V_{H1}	Input High-voltage	(Except XTAL1, RST)	$0.2 V_{CC} + 0.9$	$V_{CC} + 0.5$	V
V_{H1}	Input High-voltage	(XTAL1, RST)	$0.7 V_{CC}$	$V_{CC} + 0.5$	V
I_{OL}	Output Low-voltage ⁽¹⁾ (Ports 1,2,3)	$I_{OL} = 1.6 \text{ mA}$		0.45	V
I_{OL1}	Output Low-voltage ⁽¹⁾ (Port 0, ALE, PSEN)	$I_{OL} = 3.2 \text{ mA}$		0.45	V
V_{OH}	Output High-voltage (Ports 1,2,3, ALE, PSEN)	$I_{OH} = -60 \mu\text{A}, V_{CC} = 5V \pm 10\%$	2.4		V
		$I_{OH} = -25 \mu\text{A}$	$0.75 V_{CC}$		V
		$I_{OH} = -10 \mu\text{A}$	$0.9 V_{CC}$		V
V_{OH1}	Output High-voltage (Port 0 in External Bus Mode)	$I_{OH} = -800 \mu\text{A}, V_{CC} = 5V \pm 10\%$	2.4		V
		$I_{OH} = -300 \mu\text{A}$	$0.75 V_{CC}$		V
		$I_{OH} = -60 \mu\text{A}$	$0.9 V_{CC}$		V
I_{IL}	Logical 0 Input Current (Ports 1,2,3)	$V_{IN} = 0.45V$		-50	μA
I_{TL}	Logical 1 to 0 Transition Current (Ports 1,2,3)	$V_{IN} = 2V, V_{CC} = 5V \pm 10\%$		-650	μA
I_{LI}	Input Leakage Current (Port 0, $\bar{E}A$)	$0.45 < V_{IN} < V_{CC}$		± 10	μA
RRST	Reset Pull-down Resistor		50	300	$\text{K}\Omega$
C_{IO}	Pin Capacitance	Test Freq. = 1 MHz, $T_A = 25^\circ\text{C}$		10	pF
I_{CC}	Power Supply Current	Active Mode, 12 MHz		20	mA
		Idle Mode, 12 MHz		5	mA
	Power-down Mode ⁽²⁾	$V_{CC} = 6V$		100	μA
		$V_{CC} = 3V$		40	μA

- Notes: 1. Under steady state (non-transient) conditions, I_{OL} must be externally limited as follows:
 Maximum I_{OL} per port pin: 10 mA
 Maximum I_{OL} per 8-bit port: Port 0: 26 mA
 Ports 1, 2, 3: 15 mA
 Maximum total I_{OL} for all output pins: 71 mA
 If I_{OL} exceeds the test condition, V_{OL} may exceed the related specification. Pins are not guaranteed to sink current greater than the listed test conditions.
2. Minimum V_{CC} for Power-down is 2V.

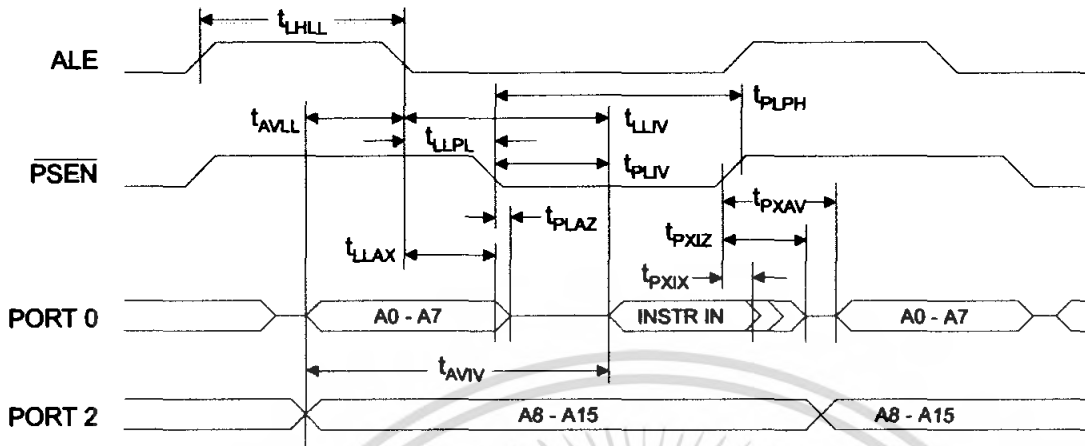
C Characteristics

Under operating conditions, load capacitance for Port 0, ALE/ $\overline{\text{PROG}}$, and $\overline{\text{PSEN}}$ = 100 pF; load capacitance for all other inputs = 80 pF.

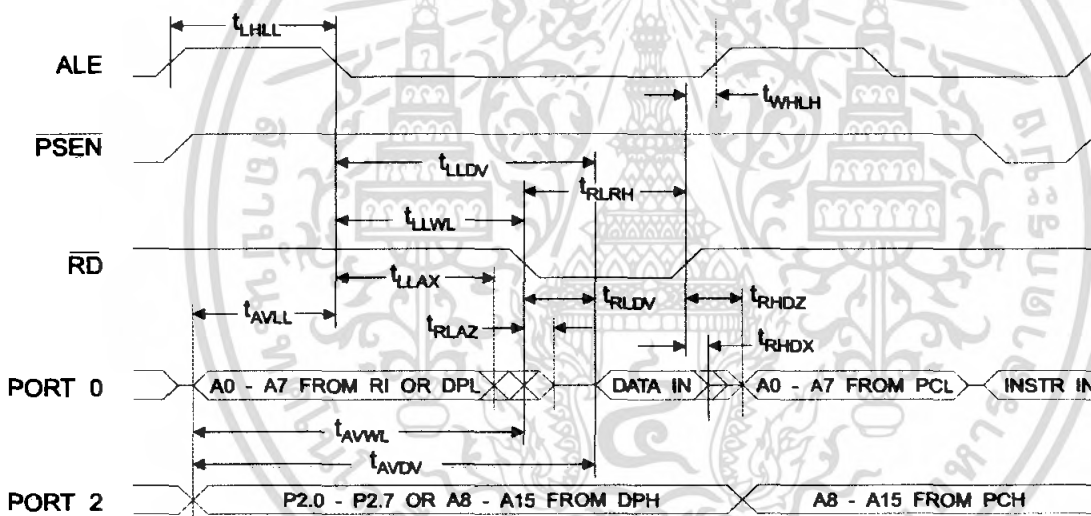
External Program and Data Memory Characteristics

Symbol	Parameter	12 MHz Oscillator		16 to 24 MHz Oscillator		Units
		Min	Max	Min	Max	
f_{CLCL}	Oscillator Frequency			0	24	MHz
t_{HLL}	ALE Pulse Width	127		$2t_{\text{CLCL}}-40$		ns
t_{VLL}	Address Valid to ALE Low	43		$t_{\text{CLCL}}-13$		ns
t_{LAX}	Address Hold after ALE Low	48		$t_{\text{CLCL}}-20$		ns
t_{LLV}	ALE Low to Valid Instruction In		233		$4t_{\text{CLCL}}-65$	ns
t_{LPL}	ALE Low to $\overline{\text{PSEN}}$ Low	43		$t_{\text{CLCL}}-13$		ns
t_{PLPH}	$\overline{\text{PSEN}}$ Pulse Width	205		$3t_{\text{CLCL}}-20$		ns
t_{PLV}	$\overline{\text{PSEN}}$ Low to Valid Instruction In		145		$3t_{\text{CLCL}}-45$	ns
t_{PXIX}	Input Instruction Hold after $\overline{\text{PSEN}}$	0		0		ns
t_{PXIZ}	Input Instruction Float after $\overline{\text{PSEN}}$		59		$t_{\text{CLCL}}-10$	ns
t_{PXAV}	$\overline{\text{PSEN}}$ to Address Valid	75		$t_{\text{CLCL}}-8$		ns
t_{AVIV}	Address to Valid Instruction In		312		$5t_{\text{CLCL}}-55$	ns
t_{PLAZ}	$\overline{\text{PSEN}}$ Low to Address Float		10		10	ns
t_{RLRH}	$\overline{\text{RD}}$ Pulse Width	400		$6t_{\text{CLCL}}-100$		ns
t_{WLWH}	$\overline{\text{WR}}$ Pulse Width	400		$6t_{\text{CLCL}}-100$		ns
t_{RLDV}	$\overline{\text{RD}}$ Low to Valid Data In		252		$5t_{\text{CLCL}}-90$	ns
t_{RHDX}	Data Hold after $\overline{\text{RD}}$	0		0		ns
t_{RHDX}	Data Float after $\overline{\text{RD}}$		97		$2t_{\text{CLCL}}-28$	ns
t_{LLDV}	ALE Low to Valid Data In		517		$8t_{\text{CLCL}}-150$	ns
t_{AVDV}	Address to Valid Data In		585		$9t_{\text{CLCL}}-165$	ns
t_{LLWL}	ALE Low to $\overline{\text{RD}}$ or $\overline{\text{WR}}$ Low	200	300	$3t_{\text{CLCL}}-50$	$3t_{\text{CLCL}}+50$	ns
t_{AVWL}	Address to $\overline{\text{RD}}$ or $\overline{\text{WR}}$ Low	203		$4t_{\text{CLCL}}-75$		ns
t_{QVWX}	Data Valid to $\overline{\text{WR}}$ Transition	23		$t_{\text{CLCL}}-20$		ns
t_{QVWH}	Data Valid to $\overline{\text{WR}}$ High	433		$7t_{\text{CLCL}}-120$		ns
t_{WHQX}	Data Hold after $\overline{\text{WR}}$	33		$t_{\text{CLCL}}-20$		ns
t_{RLAZ}	$\overline{\text{RD}}$ Low to Address Float		0		0	ns
t_{WHLH}	$\overline{\text{RD}}$ or $\overline{\text{WR}}$ High to ALE High	43	123	$t_{\text{CLCL}}-20$	$t_{\text{CLCL}}+25$	ns

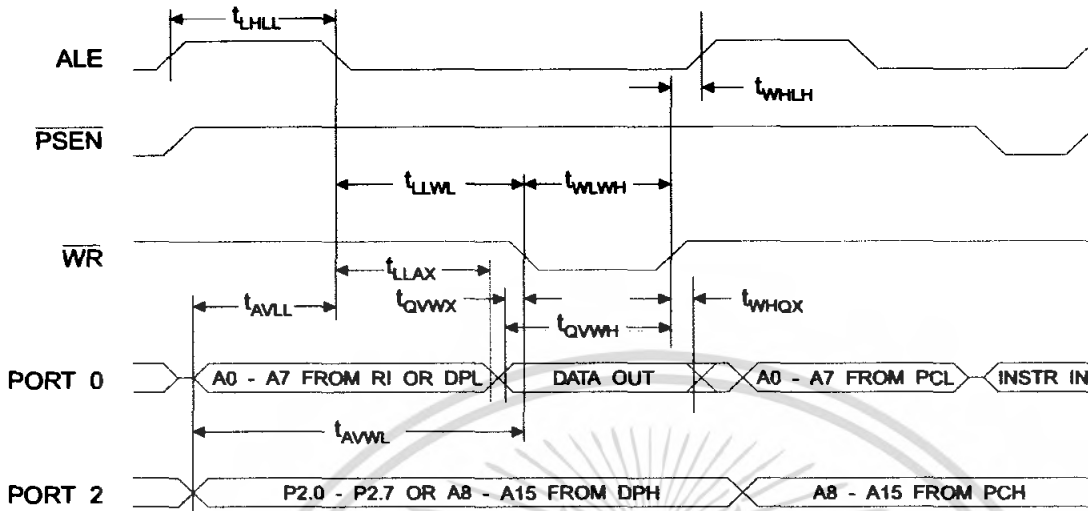
External Program Memory Read Cycle



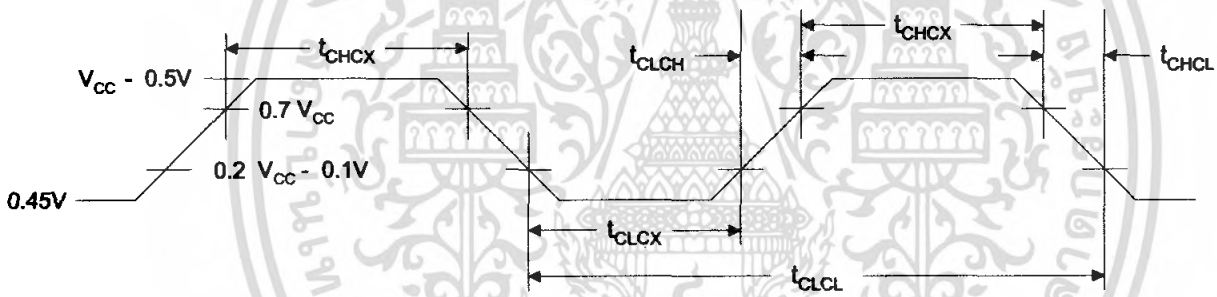
External Data Memory Read Cycle



External Data Memory Write Cycle



External Clock Drive Waveforms



External Clock Drive

Symbol	Parameter	Min	Max	Units
$1/t_{CLCL}$	Oscillator Frequency	0	24	MHz
t_{CLCL}	Clock Period	41.6		ns
t_{CHCX}	High Time	15		ns
t_{CLCX}	Low Time	15		ns
t_{CLCH}	Rise Time		20	ns
t_{CHCL}	Fall Time		20	ns

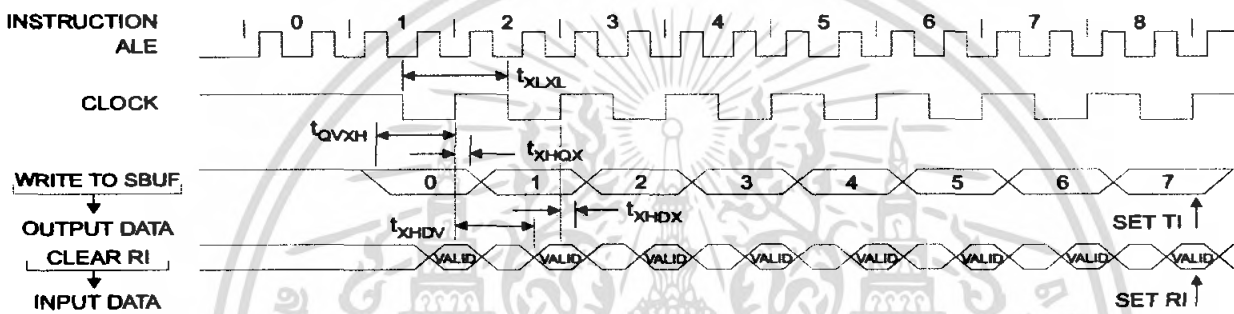


Serial Port Timing: Shift Register Mode Test Conditions

$V_{CC} = 5.0\text{ V} \pm 20\%$; Load Capacitance = 80 pF)

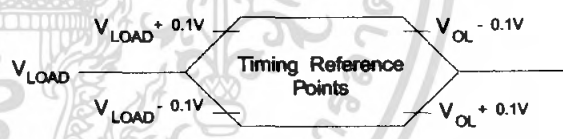
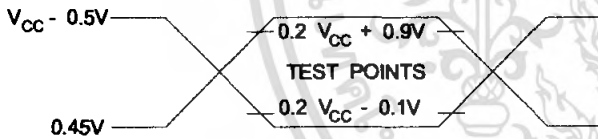
Symbol	Parameter	12 MHz Osc		Variable Oscillator		Units
		Min	Max	Min	Max	
t_{CLXL}	Serial Port Clock Cycle Time	1.0		$12t_{CLCL}$		μs
t_{QVXH}	Output Data Setup to Clock Rising Edge	700		$10t_{CLCL}-133$		ns
t_{XHDX}	Output Data Hold after Clock Rising Edge	50		$2t_{CLCL}-117$		ns
t_{XHDV}	Input Data Hold after Clock Rising Edge	0		0		ns
t_{XHDV}	Clock Rising Edge to Input Data Valid		700		$10t_{CLCL}-133$	ns

Shift Register Mode Timing Waveforms



AC Testing Input/Output Waveforms⁽¹⁾

Float Waveforms⁽¹⁾



Note: 1. AC Inputs during testing are driven at $V_{CC} - 0.5\text{V}$ for a logic 1 and 0.45V for a logic 0. Timing measurements are made at V_{TH} min. for a logic 1 and V_{IL} max. for a logic 0.

Note: 1. For timing purposes, a port pin is no longer floating when a 100 mV change from load voltage occurs. A port pin begins to float when 100 mV change from the loaded V_{OH}/V_{OL} level occurs.

Ordering Information

Speed (MHz)	Power Supply	Ordering Code	Package	Operation Range	
12	5V ±20%	AT89C51-12AC	44A	Commercial (0°C to 70°C)	
		AT89C51-12JC	44J		
		AT89C51-12PC	40P6		
		AT89C51-12QC	44Q		
			AT89C51-12AI	44A	Industrial (-40°C to 85°C)
			AT89C51-12JI	44J	
			AT89C51-12PI	40P6	
			AT89C51-12QI	44Q	
16	5V ±20%	AT89C51-16AC	44A	Commercial (0°C to 70°C)	
		AT89C51-16JC	44J		
		AT89C51-16PC	40P6		
		AT89C51-16QC	44Q		
			AT89C51-16AI	44A	Industrial (-40°C to 85°C)
			AT89C51-16JI	44J	
			AT89C51-16PI	40P6	
			AT89C51-16QI	44Q	
20	5V ±20%	AT89C51-20AC	44A	Commercial (0°C to 70°C)	
		AT89C51-20JC	44J		
		AT89C51-20PC	40P6		
		AT89C51-20QC	44Q		
			AT89C51-20AI	44A	Industrial (-40°C to 85°C)
			AT89C51-20JI	44J	
			AT89C51-20PI	40P6	
			AT89C51-20QI	44Q	
24	5V ±20%	AT89C51-24AC	44A	Commercial (0°C to 70°C)	
		AT89C51-24JC	44J		
		AT89C51-24PC	40P6		
		AT89C51-24QC	44Q		
			AT89C51-24AI	44A	Industrial (-40°C to 85°C)
			AT89C51-24JI	44J	
			AT89C51-24PI	40P6	
			AT89C51-24QI	44Q	

Package Type

44A	44-lead, Thin Plastic Gull Wing Quad Flatpack (TQFP)
44J	44-lead, Plastic J-Leaded Chip Carrier (PLCC)
40P6	40-lead, 0.600" Wide, Plastic Dual Inline Package (PDIP)
44Q	44-lead, Plastic Gull Wing Quad Flatpack (PQFP)

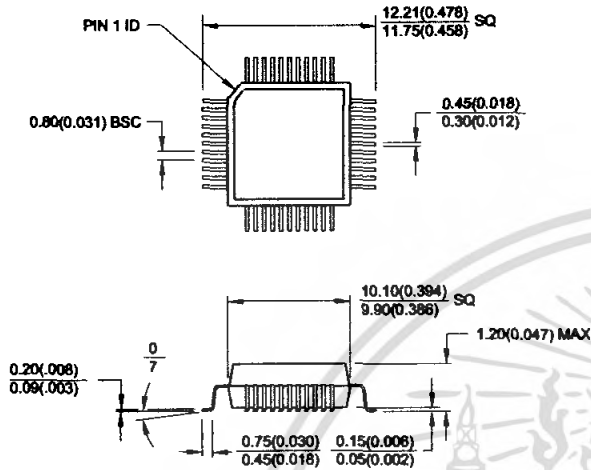


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Packaging Information

44A, 44-lead, Thin (1.0 mm) Plastic Gull Wing Quad Flatpack (TQFP)

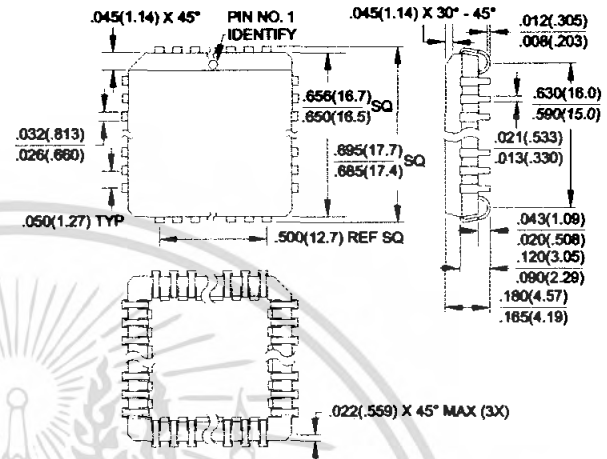
Dimensions in Millimeters and (Inches)*
JEDEC STANDARD MS-026 ACB



44J, 44-lead, Plastic J-leaded Chip Carrier (PLCC)

Dimensions in Inches and (Millimeters)

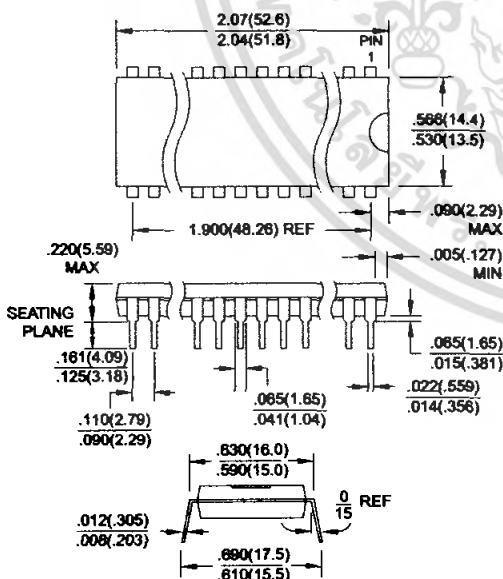
JEDEC STANDARD MS-018 AC



Controlling dimension: millimeters

40P6, 40-lead, 0.600" Wide, Plastic Dual Inline Package (PDIP)

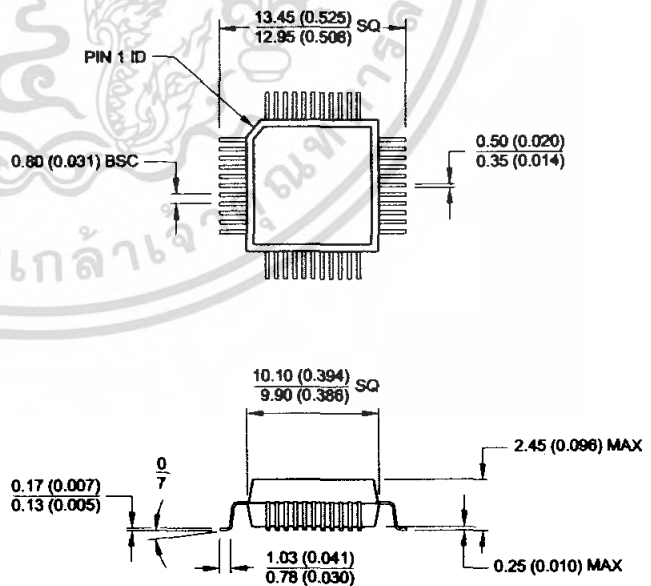
Dimensions in Inches and (Millimeters)



44Q, 44-lead, Plastic Quad Flat Package (PQFP)

Dimensions in Millimeters and (Inches)*

JEDEC STANDARD MS-022 AB



Controlling dimension: millimeters



Atmel Headquarters

Corporate Headquarters

2325 Orchard Parkway
San Jose, CA 95131
TEL (408) 441-0311
FAX (408) 487-2600

Europe

Atmel U.K., Ltd.
Coliseum Business Centre
Riverside Way
Camberley, Surrey GU15 3YL
England
TEL (44) 1276-686-677
FAX (44) 1276-686-697

Asia

Atmel Asia, Ltd.
Room 1219
Chinachem Golden Plaza
77 Mody Road Tsimhatsui
East Kowloon
Hong Kong
TEL (852) 2721-9778
FAX (852) 2722-1369

Japan

Atmel Japan K.K.
9F, Tonetsu Shinkawa Bldg.
1-24-8 Shinkawa
Chuo-ku, Tokyo 104-0033
Japan
TEL (81) 3-3523-3551
FAX (81) 3-3523-7581

Atmel Operations

Atmel Colorado Springs

1150 E. Cheyenne Mtn. Blvd.
Colorado Springs, CO 80906
TEL (719) 576-3300
FAX (719) 540-1759

Atmel Rousset

Zone Industrielle
13106 Rousset Cedex
France
TEL (33) 4-4253-6000
FAX (33) 4-4253-6001

Fax-on-Demand

North America:
1-(800) 292-8635
International:
1-(408) 441-0732

e-mail
literature@atmel.com

Web Site
<http://www.atmel.com>

BBS
1-(408) 436-4309

Atmel Corporation 2000.

Atmel Corporation makes no warranty for the use of its products, other than those expressly contained in the Company's standard warranty which is detailed in Atmel's Terms and Conditions located on the Company's web site. The Company assumes no responsibility for any errors which may appear in this document, reserves the right to change devices or specifications detailed herein at any time without notice, and does not make any commitment to update the information contained herein. No licenses to patents or other intellectual property of Atmel are granted by the Company in connection with the sale of Atmel products, expressly or by implication. Atmel's products are not authorized for use as critical components in life support devices or systems.

marks bearing ® and/or ™ are registered trademarks and trademarks of Atmel Corporation.

Terms and product names in this document may be trademarks of others.



Printed on recycled paper.

0265G-02/00/xM

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



+5V-Powered, Multichannel RS-232 Drivers/Receivers

MAX220-MAX249

General Description

The MAX220-MAX249 family of line drivers/receivers is intended for all EIA/TIA-232E and V.28/V.24 communications interfaces, particularly applications where $\pm 12V$ is not available.

These parts are especially useful in battery-powered systems, since their low-power shutdown mode reduces power dissipation to less than 5 μ W. The MAX225, MAX233, MAX235, and MAX245/MAX246/MAX247 use no external components and are recommended for applications where printed circuit board space is critical.

Next-Generation Device Features

- ◆ For Low-Voltage, Integrated ESD Applications
MAX3222E/MAX3232E/MAX3237E/MAX3241E/
MAX3246E: +3.0V to +5.5V, Low-Power, Up to
1Mbps, True RS-232 Transceivers Using Four
0.1 μ F External Capacitors (MAX3246E Available
in a UCSP™ Package)
- ◆ For Low-Cost Applications
MAX221E: $\pm 15kV$ ESD-Protected, +5V, 1 μ A,
Single RS-232 Transceiver with AutoShutdown™

Applications

- Portable Computers
- Low-Power Modems
- Interface Translation
- Battery-Powered RS-232 Systems
- Multidrop RS-232 Networks

Ordering Information

PART	TEMP RANGE	PIN-PACKAGE
MAX220CPE	0°C to +70°C	16 Plastic DIP
MAX220CSE	0°C to +70°C	16 Narrow SO
MAX220CWE	0°C to +70°C	16 Wide SO
MAX220C/D	0°C to +70°C	Dice*
MAX220EPE	-40°C to +85°C	16 Plastic DIP
MAX220ESE	-40°C to +85°C	16 Narrow SO
MAX220EWE	-40°C to +85°C	16 Wide SO
MAX220EJE	-40°C to +85°C	16 CERDIP
MAX220MJE	-55°C to +125°C	16 CERDIP

AutoShutdown and UCSP are trademarks of Maxim Integrated Products, Inc.

Ordering Information continued at end of data sheet.
*Contact factory for dice specifications.

Selection Table

Part Number	Power Supply (V)	No. of RS-232 Drivers/Rx	No. of Ext. Caps	Nominal Cap. Value (μ F)	SHDN & Three-State	Rx Active in SHDN	Data Rate (kbps)	Features
MAX220	+5	2/2	4	0.047/0.33	No	—	120	Ultra-low-power, industry-standard pinout
MAX222	+5	2/2	4	0.1	Yes	—	200	Low-power shutdown
MAX223 (MAX213)	+5	4/5	4	1.0 (0.1)	Yes	—	120	MAX241 and receivers active in shutdown
MAX225	+5	5/5	0	—	Yes	✓	120	Available in SO
MAX230 (MAX200)	+5	5/0	4	1.0 (0.1)	Yes	—	120	5 drivers with shutdown
MAX231 (MAX201)	+5 and +7.5 to +13.2	2/2	2	1.0 (0.1)	No	—	120	Standard +5/+12V or battery supplies; same functions as MAX232
MAX232 (MAX202)	+5	2/2	4	1.0 (0.1)	No	—	120 (64)	Industry standard
MAX232A	+5	2/2	4	0.1	No	—	200	Higher slew rate, small caps
MAX233 (MAX203)	+5	2/2	0	—	No	—	120	No external caps
MAX233A	+5	2/2	0	—	No	—	200	No external caps, high slew rate
MAX234 (MAX204)	+5	4/0	4	1.0 (0.1)	No	—	120	Replaces 1488
MAX235 (MAX205)	+5	5/5	0	—	Yes	—	120	No external caps
MAX236 (MAX206)	+5	4/3	4	1.0 (0.1)	Yes	—	120	Shutdown, three state
MAX237 (MAX207)	+5	5/3	4	1.0 (0.1)	No	—	120	Complements IBM PC serial port
MAX238 (MAX208)	+5	4/4	4	1.0 (0.1)	No	—	120	Replaces 1488 and 1489
MAX239 (MAX209)	+5 and +7.5 to +13.2	3/5	2	1.0 (0.1)	No	—	120	Standard +5/+12V or battery supplies; single-package solution for IBM PC serial port
MAX240	+5	5/5	4	1.0	Yes	—	120	DIP or flatpack package
MAX241 (MAX211)	+5	4/5	4	1.0 (0.1)	Yes	—	120	Complete IBM PC serial port
MAX242	+5	2/2	4	0.1	Yes	✓	200	Separate shutdown and enable
MAX243	+5	2/2	4	0.1	No	—	200	Open-line detection simplifies cabling
MAX244	+5	8/10	4	1.0	No	—	120	High slew rate
MAX245	+5	8/10	0	—	Yes	✓	120	High slew rate, int. caps, two shutdown modes
MAX246	+5	8/10	0	—	Yes	✓	120	High slew rate, int. caps, three shutdown modes
MAX247	+5	8/9	0	—	Yes	✓	120	High slew rate, int. caps, nine operating modes
MAX248	+5	8/8	4	1.0	Yes	✓	120	High slew rate, selective half-chip enables
MAX249	+5	6/10	4	1.0	Yes	✓	120	Available in quad flatpack package



Maxim Integrated Products 1

For pricing, delivery, and ordering information, please contact Maxim/Dallas Direct! at 1-888-629-4642, or visit Maxim's website at www.maxim-ic.com.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

+5V-Powered, Multichannel RS-232 Drivers/Receivers

ABSOLUTE MAXIMUM RATINGS—MAX220/222/232A/233A/242/243

Supply Voltage (V _{CC})	-0.3V to +6V	18-Pin Plastic DIP (derate 11.11mW/°C above +70°C)	889mW
V+ (Note 1)	(V _{CC} - 0.3V) to +14V	20-Pin Plastic DIP (derate 8.00mW/°C above +70°C)	440mW
V- (Note 1)	+0.3V to +14V	16-Pin Narrow SO (derate 8.70mW/°C above +70°C)	696mW
Input Voltages		16-Pin Wide SO (derate 9.52mW/°C above +70°C)	762mW
T _{IN}	-0.3V to (V _{CC} - 0.3V)	18-Pin Wide SO (derate 9.52mW/°C above +70°C)	762mW
R _{IN} (Except MAX220)	±30V	20-Pin Wide SO (derate 10.00mW/°C above +70°C)	800mW
R _{IN} (MAX220)	±25V	20-Pin SSOP (derate 8.00mW/°C above +70°C)	640mW
T _{OUT} (Except MAX220) (Note 2)	±15V	16-Pin CERDIP (derate 10.00mW/°C above +70°C)	800mW
T _{OUT} (MAX220)	±13.2V	18-Pin CERDIP (derate 10.53mW/°C above +70°C)	842mW
Output Voltages		Operating Temperature Ranges	
T _{OUT}	±15V	MAX2_AC, MAX2_C	0°C to +70°C
R _{OUT}	-0.3V to (V _{CC} + 0.3V)	MAX2_AE, MAX2_E	-40°C to +85°C
Driver/Receiver Output Short Circuited to GND	Continuous	MAX2_AM, MAX2_M	-55°C to +125°C
Continuous Power Dissipation (T _A = +70°C)		Storage Temperature Range	-65°C to +160°C
16-Pin Plastic DIP (derate 10.53mW/°C above +70°C)	842mW	Lead Temperature (soldering, 10s) (Note 3)	+300°C

Note 1: For the MAX220, V+ and V- can have a maximum magnitude of 7V, but their absolute difference cannot exceed 13V.

Note 2: Input voltage measured with T_{OUT} in high-impedance state, SHDN or V_{CC} = 0V.

Note 3: Maximum reflow temperature for the MAX233A is +225°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 in the operational sections of the specifications is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

ELECTRICAL CHARACTERISTICS—MAX220/222/232A/233A/242/243

(V_{CC} = +5V ±10%, C1–C4 = 0.1µF, MAX220, C1 = 0.047µF, C2–C4 = 0.33µF, T_A = T_{MIN} to T_{MAX}, unless otherwise noted.)

PARAMETER	CONDITIONS		MIN	TYP	MAX	UNITS
RS-232 TRANSMITTERS						
Output Voltage Swing	All transmitter outputs loaded with 3kΩ to GND		±5	±8		V
Input Logic Threshold Low				1.4	0.8	V
Input Logic Threshold High	All devices except MAX220		2	1.4		V
	MAX220: V _{CC} = 5.0V		2.4			
Logic Pullup/Input Current	All except MAX220, normal operation			5	40	µA
	SHDN = 0V, MAX222/MAX242, shutdown, MAX220			±0.01	±1	
Output Leakage Current	V _{CC} = 5.5V, SHDN = 0V, V _{OUT} = ±15V, MAX222/MAX242			±0.01	±10	µA
	V _{CC} = SHDN = 0V	V _{OUT} = ±15V		±0.01	±10	
		MAX220, V _{OUT} = ±12V			±25	
Data Rate				200	116	kbps
Transmitter Output Resistance	V _{CC} = V+ = V- = 0V, V _{OUT} = ±2V		300	10M		Ω
Output Short-Circuit Current	V _{OUT} = 0V	V _{OUT} = 0V	±7	±22		mA
		MAX220			±60	
RS-232 RECEIVERS						
RS-232 Input Voltage Operating Range					±30	V
	MAX220				±25	
RS-232 Input Threshold Low	V _{CC} = 5V	All except MAX243 R2 _{IN}	0.8	1.3		V
		MAX243 R2 _{IN} (Note 4)	-3			
RS-232 Input Threshold High	V _{CC} = 5V	All except MAX243 R2 _{IN}		1.8	2.4	V
		MAX243 R2 _{IN} (Note 4)		-0.5	-0.1	

+5V-Powered, Multichannel RS-232 Drivers/Receivers

MAX220-MAX249

ELECTRICAL CHARACTERISTICS—MAX220/222/232A/233A/242/243 (continued)

(V_{CC} = +5V ±10%, C₁-C₄ = 0.1μF, MAX220, C₁ = 0.047μF, C₂-C₄ = 0.33μF, T_A = T_{MIN} to T_{MAX}, unless otherwise noted.)

PARAMETER	CONDITIONS		MIN	TYP	MAX	UNITS
RS-232 Input Hysteresis	All except MAX220/MAX243, V _{CC} = 5V, no hysteresis in SHDN		0.2	0.5	1.0	V
	MAX220		0.3			
	MAX243		1			
RS-232 Input Resistance	T _A = +25°C (MAX220)		3	5	7	kΩ
			3	5	7	
TTL/CMOS Output Voltage Low	I _{OUT} = 3.2mA		0.2			V
	I _{OUT} = 1.6mA (MAX220)		0.4			
TTL/CMOS Output Voltage High	I _{OUT} = -1.0mA		3.5	V _{CC} - 0.2		V
TTL/CMOS Output Short-Circuit Current	Sourcing V _{OUT} = GND		-2	-10		mA
	Shrinking V _{OUT} = V _{CC}		10	30		
TTL/CMOS Output Leakage Current	SHDN = V _{CC} or EN = V _{CC} (SHDN = 0V for MAX222), 0V ≤ V _{OUT} ≤ V _{CC}		±0.05		±10	μA
EN Input Threshold Low	MAX242		1.4		0.8	V
EN Input Threshold High	MAX242		2.0		1.4	V
Operating Supply Voltage			4.5		5.5	V
V _{CC} Supply Current (SHDN = V _{CC}), figures 5, 6, 11, 19	No load	MAX220	0.5		2	μA
		MAX222/MAX232A/MAX233A/MAX242/MAX243	4		10	
	3kΩ load both inputs	MAX220	12			
		MAX222/MAX232A/MAX233A/MAX242/MAX243	15			
Shutdown Supply Current	MAX222/MAX242	T _A = +25°C	0.1		10	μA
		T _A = 0°C to +70°C	2		50	
		T _A = -40°C to +85°C	2		50	
		T _A = -55°C to +125°C	35		100	
SHDN Input Leakage Current	MAX222/MAX242				±1	μA
SHDN Threshold Low	MAX222/MAX242		1.4		0.8	V
SHDN Threshold High	MAX222/MAX242		2.0		1.4	V
Transition Slew Rate	C _L = 50pF to 2500pF, R _L = 3kΩ to 7kΩ, V _{CC} = 5V, T _A = +25°C, measured from +3V to -3V or -3V	MAX222/MAX232A/MAX233/MAX242/MAX243	6	12	30	V/μs
		MAX220	1.5	3	30.0	
Transmitter Propagation Delay TLL to RS-232 (Normal Operation), Figure 1	t _{PHLT}	MAX222/MAX232A/MAX233/MAX242/MAX243	1.3		3.5	μs
		MAX220	4		10	
	t _{PLHT}	MAX222/MAX232A/MAX233/MAX242/MAX243	1.5		3.5	
		MAX220	5		10	

Note 4: MAX243 R_{2OUT} is guaranteed to be low when R_{2IN} is ≥ 0V or is floating.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

+5V-Powered, Multichannel RS-232 Drivers/Receivers

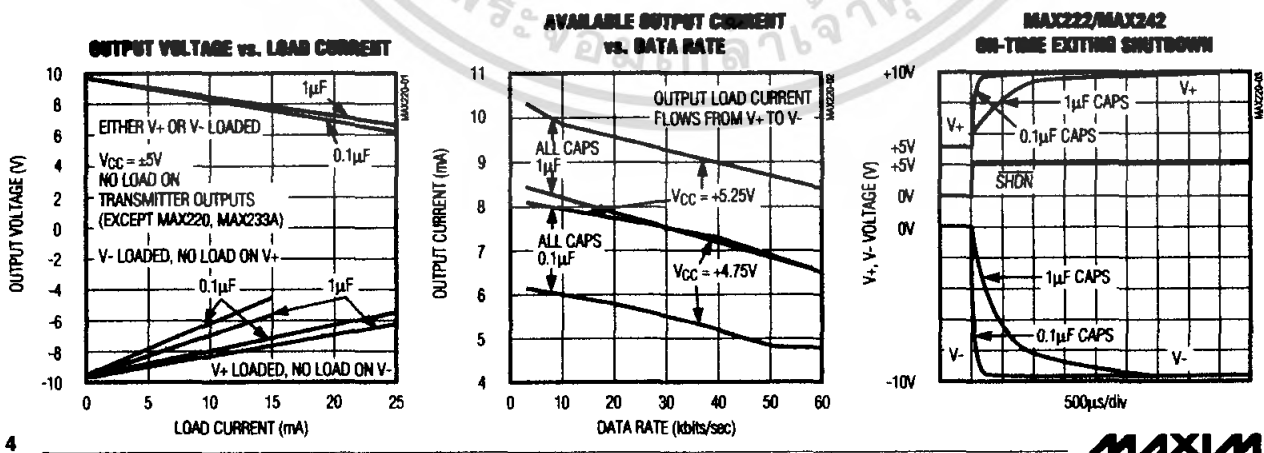
ELECTRICAL CHARACTERISTICS—MAX220/222/232A/233A/242/243 (continued)

($V_{CC} = +5V \pm 10\%$, C_1 – $C_4 = 0.1\mu F$, MAX220, $C_1 = 0.047\mu F$, C_2 – $C_4 = 0.33\mu F$, $T_A = T_{MIN}$ to T_{MAX} , unless otherwise noted.)

PARAMETER	CONDITIONS		MIN	TYP	MAX	UNITS
Receiver Propagation Delay RS-232 to TLL (Normal Operation), Figure 2	t _{PHLR}	MAX222/MAX232A/MAX233/ MAX242/MAX243		0.5	1	μs
		MAX220		0.6	3	
	t _{PLHR}	MAX222/MAX232A/MAX233/ MAX242/MAX243		0.6	1	
		MAX220		0.8	3	
Receiver Propagation Delay RS-232 to TLL (Shutdown), Figure 2	t _{PHLS}	MAX242		0.5	10	μs
	t _{PHLS}	MAX242		2.5	10	
Receiver-Output Enable Time, Figure 3	t _{ER}	MAX242		125	500	ns
Receiver-Output Disable Time, Figure 3	t _{DR}	MAX242		160	500	ns
Transmitter-Output Enable Time (SHDN Goes High), Figure 4	t _{ET}	MAX222/MAX242, 0.1μF caps (includes charge-pump start-up)		250		μs
Transmitter-Output Disable Time (SHDN Goes Low), Figure 4	t _{DT}	MAX222/MAX242, 0.1μF caps		600		ns
Transmitter + to - Propagation Delay Difference (Normal Operation)	t _{PHLT} - t _{PLHT}	MAX222/MAX232A/MAX233/ MAX242/MAX243		300		ns
		MAX220		2000		
Receiver + to - Propagation Delay Difference (Normal Operation)	t _{PHLR} - t _{PLHR}	MAX222/MAX232A/MAX233/ MAX242/MAX243		100		ns
		MAX220		225		

Typical Operating Characteristics

MAX220/MAX222/MAX232A/MAX233A/MAX242/MAX243



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

+5V-Powered, Multichannel RS-232 Drivers/Receivers

MAX220-MAX249

ABSOLUTE MAXIMUM RATINGS—MAX223/MAX230—MAX241

V _{CC}	-0.3V to +6V	20-Pin Wide SO (derate 10.00mW/°C above +70°C).....	800mW
V ₊	(V _{CC} - 0.3V) to +14V	24-Pin Wide SO (derate 11.76mW/°C above +70°C).....	941mW
V ₋	+0.3V to -14V	28-Pin Wide SO (derate 12.50mW/°C above +70°C).....	1W
Input Voltages		44-Pin Plastic FP (derate 11.11mW/°C above +70°C).....	889mW
T _{IN}	-0.3V to (V _{CC} + 0.3V)	14-Pin CERDIP (derate 9.09mW/°C above +70°C).....	727mW
R _{IN}	±30V	16-Pin CERDIP (derate 10.00mW/°C above +70°C).....	800mW
Output Voltages		20-Pin CERDIP (derate 11.11mW/°C above +70°C).....	889mW
T _{OUT}	(V ₊ + 0.3V) to (V ₋ - 0.3V)	24-Pin Narrow CERDIP	
R _{OUT}	-0.3V to (V _{CC} + 0.3V)	(derate 12.50mW/°C above +70°C).....	1W
Short-Circuit Duration, T _{OUT}	Continuous	24-Pin Sidebrazed (derate 20.0mW/°C above +70°C).....	1.6W
Continuous Power Dissipation (T _A = +70°C)		28-Pin SSOP (derate 9.52mW/°C above +70°C).....	762mW
14-Pin Plastic DIP (derate 10.00mW/°C above +70°C).....		Operating Temperature Ranges	
16-Pin Plastic DIP (derate 10.53mW/°C above +70°C).....		MAX2 __ C.....	0°C to +70°C
20-Pin Plastic DIP (derate 11.11mW/°C above +70°C).....		MAX2 __ E.....	-40°C to +85°C
24-Pin Narrow Plastic DIP		MAX2 __ M.....	-55°C to +125°C
(derate 13.33mW/°C above +70°C).....		Storage Temperature Range.....	-65°C to +160°C
1.07W		Lead Temperature (soldering, 10s) (Note 4).....	+300°C
24-Pin Plastic DIP (derate 9.09mW/°C above +70°C).....			
500mW			
16-Pin Wide SO (derate 9.52mW/°C above +70°C).....			
762mW			

Note 4: Maximum reflow temperature for the MAX233/MAX235 is +225°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 in the operational sections of the specifications is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

ELECTRICAL CHARACTERISTICS—MAX223/MAX230—MAX241

(MAX223/230/232/234/236/237/238/240/241, V_{CC} = +5V ±10%; MAX233/MAX235, V_{CC} = 5V ±5%, C₁-C₄ = 1.0μF; MAX231/MAX239, V_{CC} = 5V ±10%; V₊ = 7.5V to 13.2V; T_A = T_{MIN} to T_{MAX}; unless otherwise noted.)

PARAMETER	CONDITIONS	MIN	TYP	MAX	UNITS
Output Voltage Swing	All transmitter outputs loaded with 3kΩ to ground	±5.0	±7.3		V
V _{CC} Power-Supply Current	No load, T _A = +25°C	MAX232/233	5	10	mA
		MAX223/230/234-236/240/241	7	15	
		MAX231/239	0.4	1	
V ₊ Power-Supply Current		MAX231	1.8	5	mA
		MAX239	5	15	
Shutdown Supply Current	T _A = +25°C	MAX223	15	50	μA
		MAX230/235/236/240/241	1	10	
Input Logic Threshold Low	T _{IN} : EN, SHDN (MAX233); EN, SHDN (MAX230/235-241)			0.8	V
Input Logic Threshold High	T _{IN}	2.0			V
	EN, SHDN (MAX223); EN, SHDN (MAX230/235/236/240/241)	2.4			
Logic Pull-Up Current	T _{IN} = 0V		1.5	200	μA
Receiver Input Voltage Operating Range		-30		+30	V

+5V-Powered, Multichannel RS-232 Drivers/Receivers

ELECTRICAL CHARACTERISTICS—MAX223/MAX230—MAX241 (continued)

(MAX223/230/232/234/236/237/238/240/241, $V_{CC} = +5V \pm 10\%$; MAX233/MAX235, $V_{CC} = 5V \pm 5\%$, $C1-C4 = 1.0\mu F$; MAX231/MAX239, $V_{CC} = 5V \pm 10\%$; $V+ = 7.5V$ to $13.2V$; $T_A = T_{MIN}$ to T_{MAX} ; unless otherwise noted.)

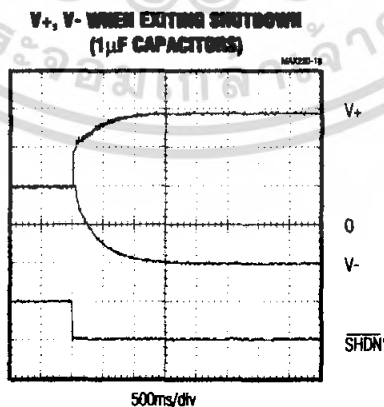
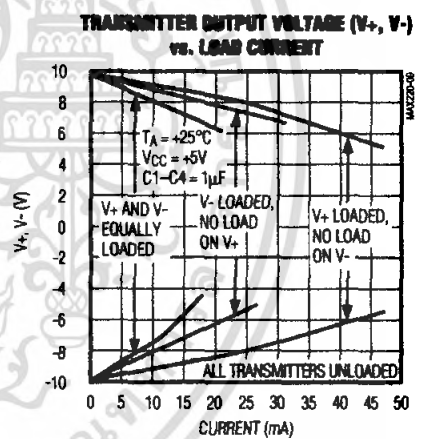
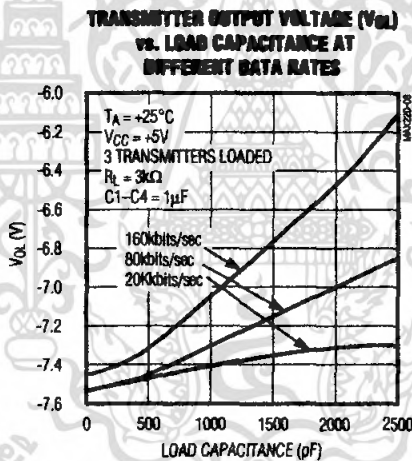
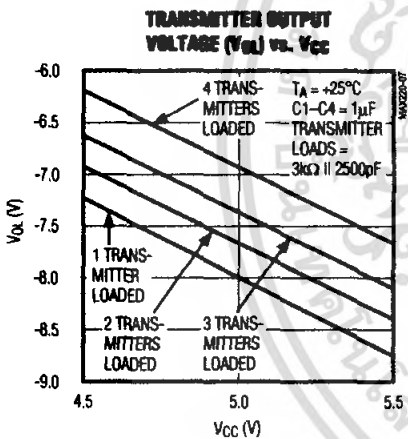
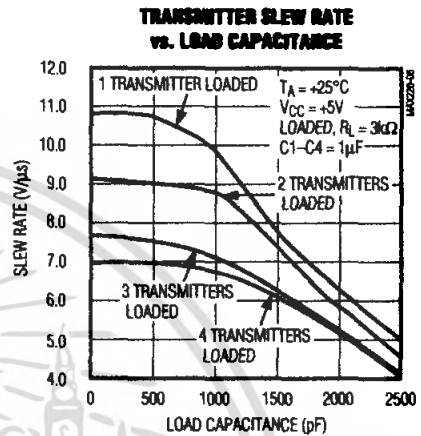
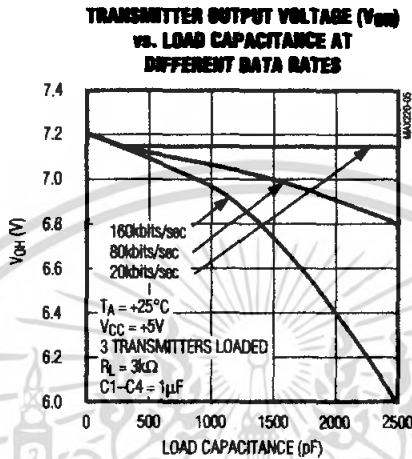
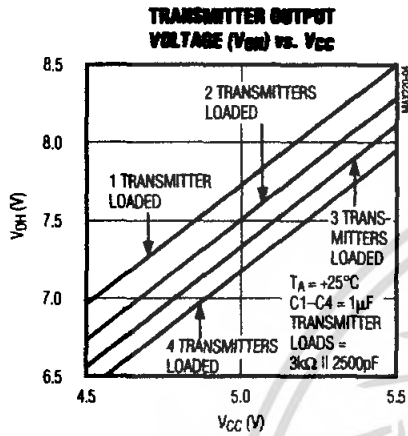
PARAMETER	CONDITIONS		MIN	TYP	MAX	UNITS
RS-232 Input Threshold Low	$T_A = +25^\circ C$, $V_{CC} = 5V$	Normal operation SHDN = 5V (MAX223) SHDN = 0V (MAX235/236/240/241)	0.8	1.2		V
		Shutdown (MAX223) SHDN = 0V, EN = 5V (R_{4IN} , R_{5IN})	0.6	1.5		
RS-232 Input Threshold High	$T_A = +25^\circ C$, $V_{CC} = 5V$	Normal operation SHDN = 5V (MAX223) SHDN = 0V (MAX235/236/240/241)		1.7	2.4	V
		Shutdown (MAX223) SHDN = 0V, EN = 5V (R_{4IN} , R_{5IN})		1.5	2.4	
RS-232 Input Hysteresis	$V_{CC} = 5V$, no hysteresis in shutdown		0.2	0.5	1.0	V
RS-232 Input Resistance	$T_A = +25^\circ C$, $V_{CC} = 5V$		3	5	7	k Ω
TTL/CMOS Output Voltage Low	$I_{OUT} = 1.6mA$ (MAX231/232/233, $I_{OUT} = 3.2mA$)				0.4	V
TTL/CMOS Output Voltage High	$I_{OUT} = -1mA$		3.5	$V_{CC} - 0.4$		V
TTL/CMOS Output Leakage Current	$0V \leq R_{OUT} \leq V_{CC}$; EN = 0V (MAX223); EN = V_{CC} (MAX235-241)			0.05	± 10	μA
Receiver Output Enable Time	Normal operation	MAX223		600		ns
		MAX235/236/239/240/241		400		
Receiver Output Disable Time	Normal operation	MAX223		900		ns
		MAX235/236/239/240/241		250		
Propagation Delay	RS-232 IN to TTL/CMOS OUT, $C_L = 150pF$	Normal operation		0.5	10	μs
		SHDN = 0V (MAX223)	t_{PHLS}	4	40	
			t_{PLHS}	6	40	
Transition Region Slew Rate	MAX223/MAX230/MAX234-241, $T_A = +25^\circ C$, $V_{CC} = 5V$, $R_L = 3k\Omega$ to $7k\Omega$, $C_L = 50pF$ to $2500pF$, measured from $+3V$ to $-3V$ or $-3V$ to $+3V$		3	5.1	30	V/ μs
	MAX231/MAX232/MAX233, $T_A = +25^\circ C$, $V_{CC} = 5V$, $R_L = 3k\Omega$ to $7k\Omega$, $C_L = 50pF$ to $2500pF$, measured from $+3V$ to $-3V$ or $-3V$ to $+3V$			4	30	
Transmitter Output Resistance	$V_{CC} = V+ = V- = 0V$, $V_{OUT} = \pm 2V$		300			Ω
Transmitter Output Short-Circuit Current				± 10		mA

+5V-Powered, Multichannel RS-232 Drivers/Receivers

MAX220-MAX249

Typical Operating Characteristics

MAX223/MAX230-MAX241



*SHUTDOWN POLARITY IS REVERSED FOR NON MAX241 PARTS

+5V-Powered, Multichannel RS-232 Drivers/Receivers

MAX220-MAX249

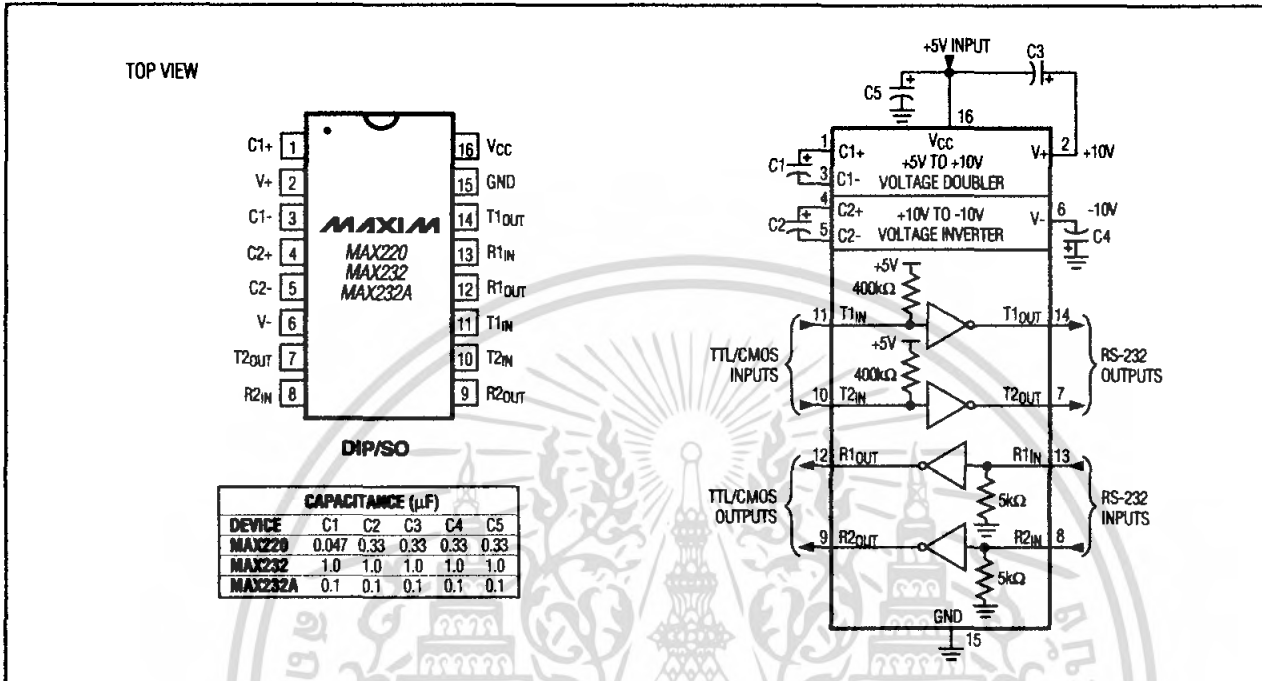


Figure 5. MAX220/MAX232/MAX232A Pin Configuration and Typical Operating Circuit

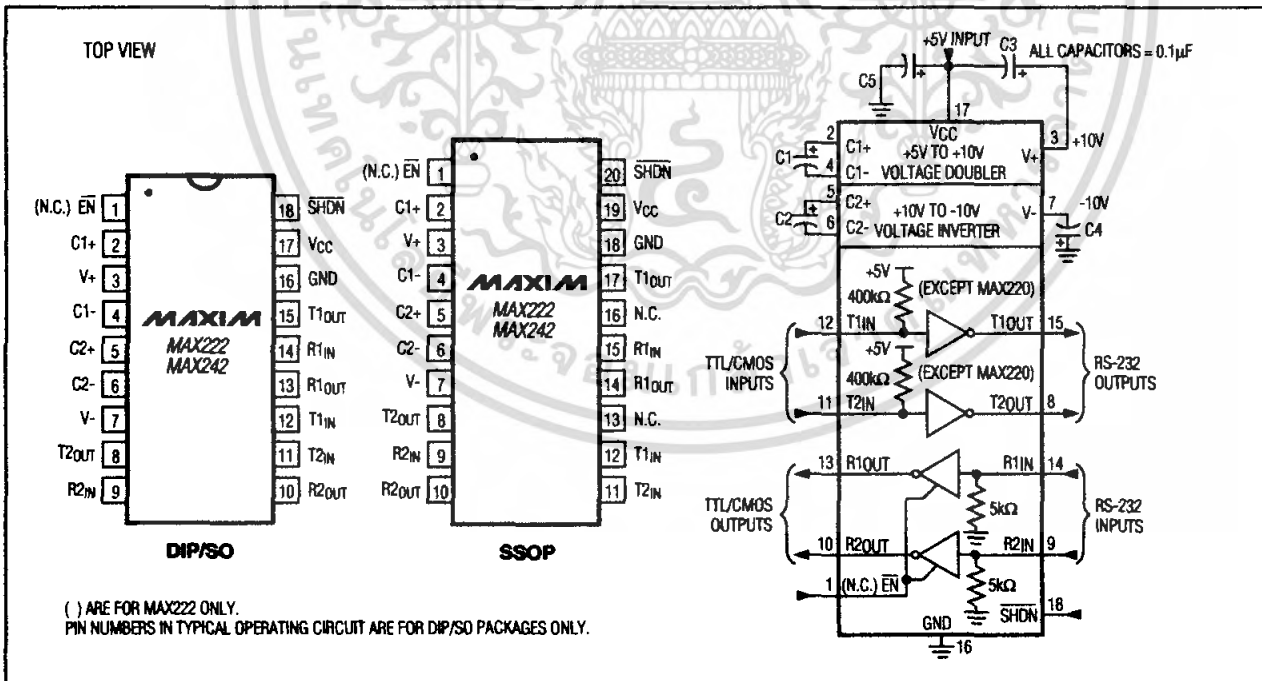


Figure 6. MAX222/MAX242 Pin Configurations and Typical Operating Circuit

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้