

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

เครื่องจำลองโครงสร้าง 3 มิติ
3D SCANNER



โดย

นางสาวหทัย คงมาก

นายอรรณพ บุญทวีวัฒน์

นางสาวอรุณี เอี่ยมปรีชา

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรสำหรับปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาอิเล็กทรอนิกส์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2541

9 17
2541

เลขหมู่.....
เลขทะเบียน 34066
วัน, เดือน, ปี 1 ต.ค. 2542

เอกสารนี้จัดทำขึ้นเพื่อการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ในการแสวงหาผลประโยชน์ และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เครื่องจำลองโครงสร้าง 3 มิติ

3D SCANNER

โดย

นางสาวหทัย คงมาก

นายอรรถพล บุญทวีพัฒน์

นางสาวอรุณี เอี่ยมปรีชา

อาจารย์ที่ปรึกษา

อาจารย์สุรเดช ศรีไตรลักษณ์

ปริญญานิพนธ์สำหรับปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2541

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาโทปีการศึกษา 2541

ภาควิชาอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง เครื่องจำลองโครงสร้าง 3 มิติ

ผู้จัดทำ

นางสาวหฤทัย คงมาก

นายอรรถพล บุญทวีพัฒน์

นางสาวอรุณี เข็มปรีชา



(อาจารย์สุรเดช ศรีไตรลักษณ์)

อาจารย์ที่ปรึกษา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เครื่องจำลอง โครงสร้าง 3 มิติ

3D SCANNER

นางสาวหฤทัย คงมาก รหัส 38014610

นายอรรถพล บุญทวีพัฒน์ รหัส 38014625

นางสาวอรุณี เขี่ยมปรีชา รหัส 38014635

โครงการได้รับการตรวจสอบแล้ว พร้อมทั้งท่านจะทำการตรวจสอบได้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ในขณะที่ผู้จัดทำโครงการชิ้นนี้ขึ้นมาได้ประสบกับปัญหาต่างๆมากมายที่เกิดขึ้น แต่แล้วก็สามารถแก้ไขปัญหานั้นมาได้ ซึ่งก็ด้วยความช่วยเหลือจากหลายๆท่านด้วยกัน ไม่ว่าจะเป็นเพื่อนๆที่ช่วยรับฟังปัญหาแล้วช่วยให้แนวคิดในมุมมองที่ต่างออกไป เพื่อนๆภาคเครื่องกลที่ช่วยให้คำปรึกษาเกี่ยวกับงานด้านฮาร์ดแวร์ ทั้งยังให้ความเอื้อเฟื้อทั้งทางด้านอุปกรณ์และสถานที่อีกด้วย และโดยเฉพาะอย่างยิ่งอาจารย์ที่ปรึกษาซึ่งให้คำปรึกษาและความสนใจเป็นอย่างดี ต้องขอขอบคุณผู้มีพระคุณทุกท่านที่ทำให้โครงการชิ้นนี้สำเร็จลุล่วงไปด้วยดี

(นางสาวหฤทัย คงมาก)

(นายอรรถพร บุญทวีพัฒน์)

(นางสาวอรุณี เอี่ยมปรีชา)

ผู้จัดทำ

เครื่องจำลองโครงสร้าง 3 มิติ

นางสาว หฤทัย คงมาก

นายอรรถพร บุญทวีวัฒน์

นางสาวอรุณี เขียมปรีชา

อ.สุรเดช ศรีไตรลักษณ์(อาจารย์ที่ปรึกษา)

ปีการศึกษา 2541

บทคัดย่อ

เครื่องจำลองโครงสร้าง 3 มิติเป็นเครื่องมือที่ใช้ในการเก็บข้อมูลเกี่ยวกับรูปลักษณะของวัตถุแล้วประมวลผลออกมาเป็นภาพ 3 มิติ โดยทำการถ่ายภาพของวัตถุในแต่ละมุมจนครบ หนึ่งรอบของวัตถุ โดยใช้กล้องดิจิทัล และแสงเลเซอร์ในการถ่ายภาพ ซึ่งในแต่ละภาพนั้นจะให้ระยะของจุดเลเซอร์ถึงจุดศูนย์กลางของฐานหมุน จากนั้นเลื่อนตำแหน่งของเลเซอร์ให้สูงขึ้น แล้วทำการถ่ายภาพในแต่ละมุมจนครบ 360 องศา จะได้ภาพในแนวนอนของวัตถุ ณ ความสูงต่างๆ นำภาพตัดขวางที่ได้ไปประมวลผลประกอบกันเป็นภาพ 3 มิติของวัตถุ แล้วแสดงออกทางหน้าจอคอมพิวเตอร์ โดยที่สามารถเก็บข้อมูลที่นำไปสร้างภาพ 3 มิติให้อยู่ในแฟ้มข้อมูลได้ อีกทั้งสามารถทำการหมุนภาพ เปลี่ยนจุดศูนย์กลางการหมุน และย่อหรือขยายขนาดของภาพได้

3 D SCANNER

Miss Haruthai Kongmark

Mr Anop Boonthaveepath

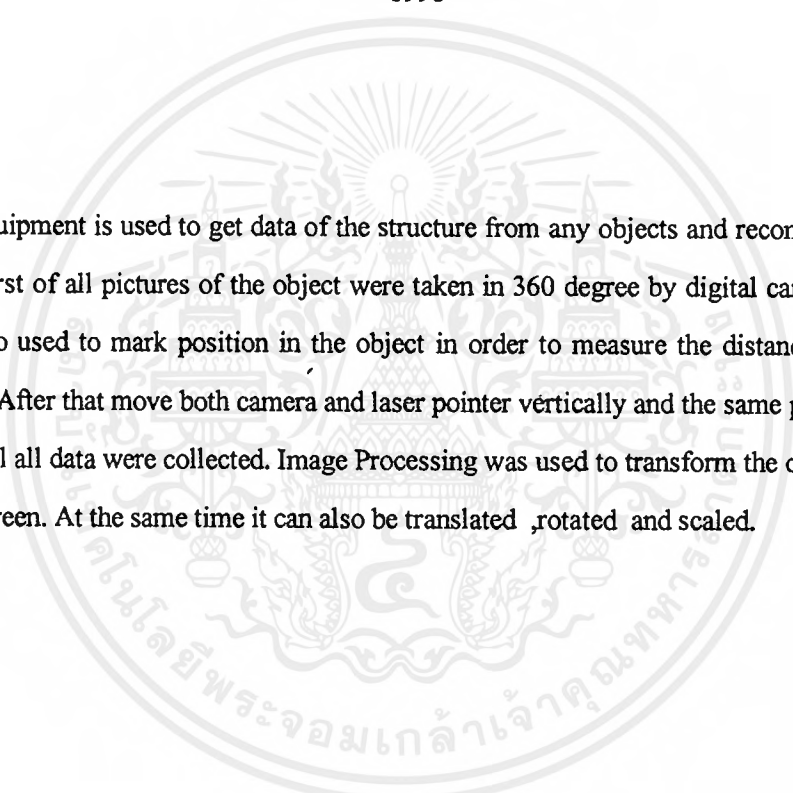
Miss Arunee Eiampreecha

Mr.Suradate Treetrailuksana(Advisor)

1998

ABSTRACT

This equipment is used to get data of the structure from any objects and reconstruct them to 3D figure. First of all pictures of the object were taken in 360 degree by digital camera. Laser pointer was also used to mark position in the object in order to measure the distance from the center of base. After that move both camera and laser pointer vertically and the same process was done again ,until all data were collected. Image Processing was used to transform the data into 3D image on the screen. At the same time it can also be translated ,rotated and scaled.



สารบัญ

	หน้า
กิตติกรรมประกาศ	I
บทคัดย่อ	II
Abstract	III
สารบัญ	IV
สารบัญรูป	VI
บทที่ 1 บทนำ	1
1.1 วัตถุประสงค์	1
1.2 ขอบเขตการทำงาน	1
1.3 ผลที่คาดว่าจะได้รับ	2
บทที่ 2 แนวคิดและหลักการ 3D SCANNER	3
2.1 การเก็บข้อมูลของวัตถุเชิง 3 มิติ	5
2.2 การเรียงพารามิเตอร์ทั้ง 3 กลับไปเป็นรูปทรง 3 มิติ	8
2.3 แนวความคิดในการเก็บข้อมูลที่จะใช้เป็นอัลกอริธึม	8
บทที่ 3 ทฤษฎีพื้นฐานและหลักการทำงานของสเตปมอเตอร์	12
3.1 คุณสมบัติทั่วไป	12
3.2 หลักการทำงานของสเตปมอเตอร์	13
3.3 การ Control Step Motor	16
บทที่ 4 กล้อง Color Video Camera	18
4.1 สิ่งที่ต้องมีในการใช้กล้อง Color Video Camera	19
4.2 คุณสมบัติพิเศษของกล้อง Color Video Camera	19
บทที่ 5 คณิตศาสตร์ในการสร้างโครงสร้าง 3 มิติ	20
5.1 Two-Dimensional Coordinate Geometry	20
5.2 Three-Dimensional Coordinate Geometry	23
บทที่ 6 ชุดทดลองค้นแบบ	28
6.1 ส่วนประกอบของชุดทดลอง	28
6.2 การขับมอเตอร์	29
บทที่ 7 โปรแกรมการทำงาน	31
7.1 โครงสร้างการทำงานของกล้องถ่ายภาพ 3 มิติ	31

	หน้า
7.2 โครงสร้างของโปรแกรมการทำงาน	31
บทที่ 8 การทดลองและผลการทดลอง	35
8.1 วิธีทดลอง	35
8.2 ผลการทดลอง	35
บทที่ 9 สรุปผลและวิจารณ์	43
9.1 สรุปผลการทดลอง	43
9.2 วิจารณ์ผลการทดลอง	43
ภาคผนวก	
เอกสารอ้างอิง	



สารบัญรูป

	หน้า
บทที่ 2	
รูปที่ 2.1 แสดง wafer แต่ละชั้นของปรีสมิค	3
รูปที่ 2.2 แสดง wafer แต่ละชั้นของทรงกระบอก	4
รูปที่ 2.3 แสดง Top View ของทรงกระบอกและ wafer	4
รูปที่ 2.4 แสดง Top View ของปรีสมิคและ wafer	5
รูปที่ 2.5 การสร้าง wafer จากวัสดุ และการนำ wafer กลับมาสร้างเป็นวัสดุ	6
รูปที่ 2.6 แสดงจุดอ้างอิงของงาน หมุนและจุดบนเส้นรอบรูปของวัสดุ	6
รูปที่ 2.7 แสดงจุด 2 จุดบนเส้นรอบรูปของวัสดุทำมุมระหว่างกัน	7
รูปที่ 2.8 แสดงความแตกต่างระหว่างเส้นรอบรูปที่ได้จากการใช้ จุด 4 จุด และ 8 จุดต่อชั้น	7
รูปที่ 2.9 แสดงการนำจุดที่ได้มาเรียงกัน โดยแต่ละจุดทำมุมกัน 360/ จำนวนจุดต่อชั้น	8
รูปที่ 2.10 แต่ละชั้นมีจุดอ้างอิงเดียวกัน	8
รูปที่ 2.11 แสดง Top View และ Side View ของงานหมุน	9
รูปที่ 2.12 ยึดคล้องกับเลเซอร์โดยเอา wafer ออกทีละชั้น	10
รูปที่ 2.13 เลื่อนคล้องกับเลเซอร์ขึ้นพร้อมกันในการทำ wafer แต่ละชั้น	10
รูปที่ 2.14 แสดงจุดอ้างอิงบน wafer และ งานหมุน	11
บทที่ 3	
รูปที่ 3.1 แสดงโครงสร้างของไฮบริดจ์สเตปมอเตอร์ที่มีจำนวน สเตปต่อรอบเท่ากับ 12	13
รูปที่ 3.2 แสดงขั้นตอนการทำงานของสเตปมอเตอร์แบบเต็มสเตป 1 เฟส	14
รูปที่ 3.3 แสดงขั้นตอนการทำงานของสเตปมอเตอร์แบบเต็มสเตป 2 เฟส	15
รูปที่ 3.4 แสดงขั้นตอนการทำงานของสเตปมอเตอร์แบบครึ่งสเตป	15

	หน้า
บทที่ 4	
รูปที่ 4.1 แสดงลักษณะของกล้อง Color Video Camera	18
รูปที่ 4.2 แสดงการต่อกล้องเข้ากับเครื่อง PC	18
บทที่ 6	
รูปที่ 6.1 ชุดทดลองค้นแบบ	28
รูปที่ 6.2 แสดงบล็อกไดอะแกรมแสดงการควบคุมการทำงานของสเตปมอเตอร์	29
รูปที่ 6.3 แสดงบล็อกไดอะแกรมของวงจรไครฟี่มอเตอร์	29
รูปที่ 6.4 แสดงวงจรไครฟี่มอเตอร์	30
บทที่ 7	
รูปที่ 7.1 แสดงการทำงานของการทำงานของการจำลองภาพ 3 มิติ	31
รูปที่ 7.2 แสดง flow chart ของไครฟี่มอเตอร์	32
รูปที่ 7.3 แสดง flow chart ของโปรแกรมประมวลผล	33
รูปที่ 7.4 แสดง flow chart ของโปรแกรมสร้างภาพ 3 มิติ	34
บทที่ 8	
รูปที่ 8.1 แสดงหน้าจอรับข้อมูล	35
รูปที่ 8.2 แสดงหน้าจอแสดงผล	36
รูปที่ 8.3 แสดงภาพที่ได้จากการสแกนด้วยที่ 25,50,100,200 step/round	37
รูปที่ 8.4 แสดงภาพที่ได้จากการสแกนปริซึมฐานหกเหลี่ยม ที่ 25,50,100,200 step/round	37
รูปที่ 8.5 แสดงภาพที่ได้จากการสแกนปริซึมฐานสี่เหลี่ยมผืนผ้า ที่ 25,50,100,200 step/round	38
รูปที่ 8.6 แสดงภาพที่ได้จากการสแกนปริซึมฐานรูปดาว ที่ 200 step/round 1 และ 3 ชั้น	38
รูปที่ 8.7 แสดงภาพที่ได้จากการสแกนปริซึมฐานสี่เหลี่ยมผืนผ้า ที่ 100 step/round 5 ชั้น	39

	หน้า
รูปที่ 8.8 (a) แสดงภาพที่ได้จากการสแกนปีระมิดฐานสี่เหลี่ยม ที่ 200 step/round 9 ชั้น	
(b) แสดงภาพที่ได้จากการสแกนปีระมิดฐานสี่เหลี่ยม ที่นำจุดผิดที่เกิดจากแสง ไม่ตกกระทบผิววัตถุออก	39
รูปที่ 8.9 แสดงภาพที่ได้จากการสแกนปีระมิดฐานสี่เหลี่ยม ที่นำจุดผิดที่เกิดจากเมล็ดสีของกล้องออก	40
รูปที่ 8.10 แสดงภาพที่ได้จากการสแกนปีระมิดฐานสี่เหลี่ยมที่สมบูรณ์	40
รูปที่ 8.11 แสดงภาพที่ได้จากการสแกนแก้วกาแฟที่ 50 step/round 41 13 ชั้น	



บทที่ 1

บทนำ

ในปัจจุบันนี้ การประยุกต์การใช้งานรูปภาพในเชิง 3 มิติ ได้มีการประยุกต์ไปในหลายแขนงเช่นด้านการแพทย์ ตัวอย่างเช่น ระบบ MRI (Magnetic Resonance Image) ที่ใช้ในการสแกนอวัยวะในร่างกาย อีกตัวอย่างหนึ่งที่สำคัญก็คือ การทำ Animation ในภาพยนตร์ การประมวลผลภาพถ่ายดาวเทียมที่กำลังนำการประมวลผลเชิง 3 มิติมาใช้ ระบบ Visual ต่างๆ ก็ต้องการการสร้าง การวิเคราะห์ และการสังเคราะห์ ในเชิง 3 มิติ อย่างมาก และระบบ 3 มิติยังจำเป็นสำหรับการประยุกต์ใช้งานในสาขา Computer Vision อีกด้วย ดังตัวอย่างที่กล่าวขึ้นมาพบว่า ระบบ 3 มิติ เป็นสิ่งที่ไม่ใช่เพื่อฝันอีกต่อไป และจะไม่ใช้ระบบที่จะอยู่เฉพาะในห้องทดลองเท่านั้น

อีกสิ่งหนึ่งที่สนับสนุน ระบบ 3 มิติมากขึ้นก็คือ คอมพิวเตอร์ที่มีศักยภาพมากขึ้น และมีความเร็วมากขึ้น และราคายังถูกลง ตลอดจนระบบ Internet ยังสนับสนุนระบบ 3 มิติมากขึ้น ระบบ 3 มิติจึงได้มีการศึกษาและประยุกต์ใช้ในหลายแขนงมากมายและมีอุปกรณ์ประยุกต์ใช้มากขึ้น ตัวอย่างหนึ่งของอุปกรณ์ที่เกี่ยวกับระบบ 3 มิติ คือ Scanner

โครงการนี้ จะเกี่ยวกับการสร้าง ระบบ Scanner ในเชิง 3 มิติ โดยใช้สาขาวิชาทาง Computer Vision ในการสร้างอุปกรณ์ชิ้นนี้

1.1 วัตถุประสงค์

โครงการนี้เราจะสร้างอุปกรณ์ที่ใช้ในการเก็บข้อมูลในเชิง 3 มิติ เพื่อที่จะสร้างโครงร่าง 3 มิติ แบบที่เหมาะสมในการใช้งานด้าน Image Processing และทางด้าน Computer Vision เช่นการจดจำรูปวัตถุในเชิง 3 มิติ การเปลี่ยนรูปของวัตถุ

1.2 ขอบเขตการทำงาน

โครงการนี้จะพยายามสร้าง 3D Scanner โดยจะเน้นที่โครงสร้างของวัตถุ มากกว่ารายละเอียดของวัตถุ โดยจะหาวัตถุที่นำมาสแกนที่เหมาะสมคือ วัตถุรูปทรงเรขาคณิต เช่น ทรงกระบอก ทรงกลม ก้อน พีรามิด หรืออาจจะเป็นวัตถุที่ใช้ในชีวิตประจำวัน เช่น แก้ว ขาม ขวด กระป๋องอมสิน

1.3 ผลที่คาดว่าจะได้รับ

จากผลการทดลองที่ได้รับในเทอมที่แล้ว แสดงให้เห็นว่าแนวความคิดเป็นจริงในเทอมนี้ ผลที่ได้รับคือผลที่ต่อเนื่องจากเทอมที่แล้ว กล่าวคือ สามารถเก็บข้อมูลในเชิง 3 มิติ ของโครงสร้างของวัตถุที่นำมา สแกน และสามารถแสดงรูป 3 มิติ ที่ได้จากการ สแกน รวมทั้งสามารถทำการเปลี่ยนจุดศูนย์กลาง (Translation) , การหมุนวัตถุ (Rotation) , การขยายและลดขนาดวัตถุ (Scaling) ซึ่งทั้ง 3 กระบวนการขั้นต้นนี้สามารถทำได้ในทุกแกน ดังนั้นเราจึงสามารถได้รูปภาพที่แสดงในจอแสดงผลในทุกมุมมองของวัตถุจริง และระบบการทำงานทั้งหมดเป็นอัตโนมัติทั้งหมด (กดปุ่มเดียวสามารถทำงานตั้งแต่ต้นจนจบ แสดงเป็นภาพขึ้นมาได้) และผลที่ได้ต้องออกมาเหมือนจริงให้มากที่สุดเท่าที่จะเป็นไปได้

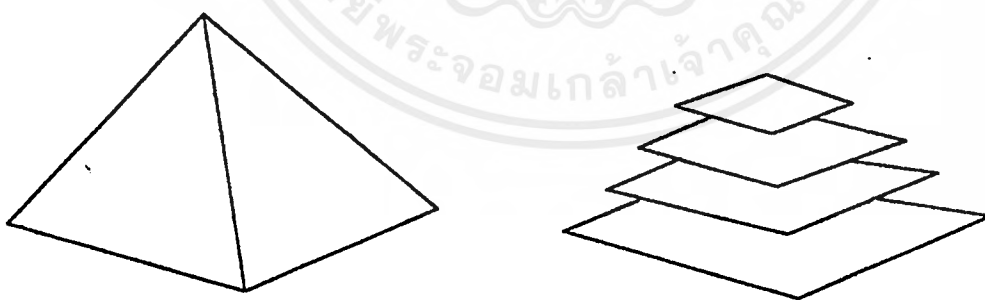


บทที่ 2

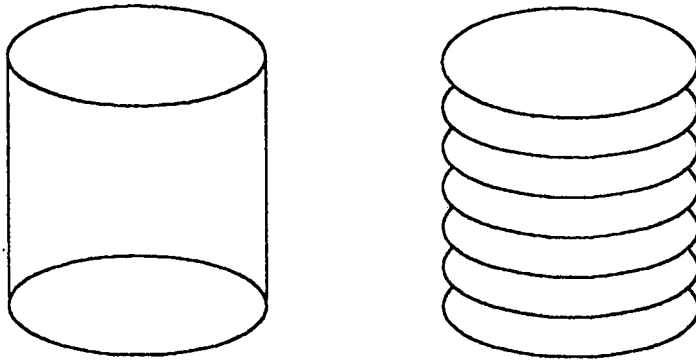
แนวคิดและหลักการของ 3D SCANNER

เมื่อเราต้องการสร้างภาพของวัตถุที่นำมาสแกน ให้แสดงผลทางหน้าจอในเชิง 3 มิติ นั้น แนวคิดเบื้องต้นของการสร้างนี้คือ ถ้าเราสามารถนำวัตถุจริงมาหันทางด้านความสูง ให้มีความหนาอย่างมาก จนสามารถมองได้ว่า เป็นแผ่นบางมากๆ จะขอเรียกแผ่นที่มีความบางมากนี้ว่า wafer เพราะความบางของ wafer นี้เราพูดได้ว่าเป็นรูปทรง 2 มิติ (ไม่มีมิติความสูง) และสามารถเก็บเส้นรอบรูปของ wafer นี้ได้ ดังนั้นจะสามารถนำ wafer ที่ได้นี้กลับมาสร้างภาพในเชิง 3 มิติได้ โดยการนำ wafer ที่ถูกหันมาวางซ้อนทับกันตามลำดับชั้นที่ได้จากการสแกน เราก็จะได้รูปวัตถุเชิง 3 มิติ

เนื่องจาก wafer นั้นไม่มีความหนา ในความเป็นจริงนั้นเราไม่สามารถสร้าง wafer ที่ไม่มีความหนาเลยได้ เราสามารถสร้างได้เพียง wafer ที่มีความหนาอย่างมากเท่านั้น ซึ่งอาจจะมีความสูง 1 mm , 2 mm หรืออาจมากกว่านี้ก็ได้ แต่ถ้า wafer ยิ่งมีความหนามาก ข้อมูลที่ได้ก็จะเป็นข้อมูลที่ไม่ต่อเนื่อง แต่ถ้าสามารถสร้าง wafer ที่บางมากขึ้น ข้อมูลที่ได้ก็จะต่อเนื่องมากขึ้น



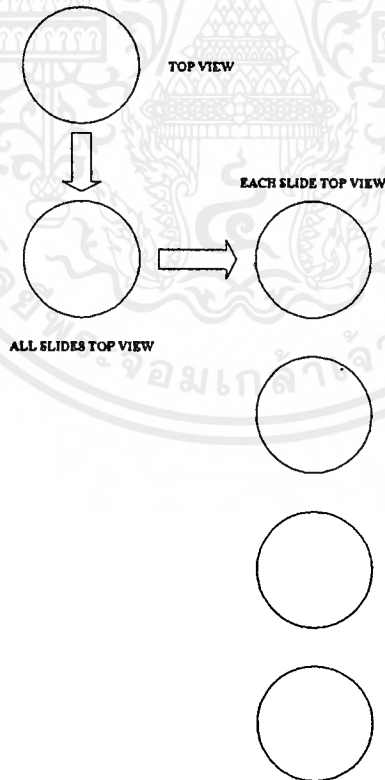
รูปที่ 2.1 แสดง wafer แต่ละชั้นของปริมาตร



รูปที่ 2.2 แสดง wafer แต่ละชั้นของทรงกระบอก

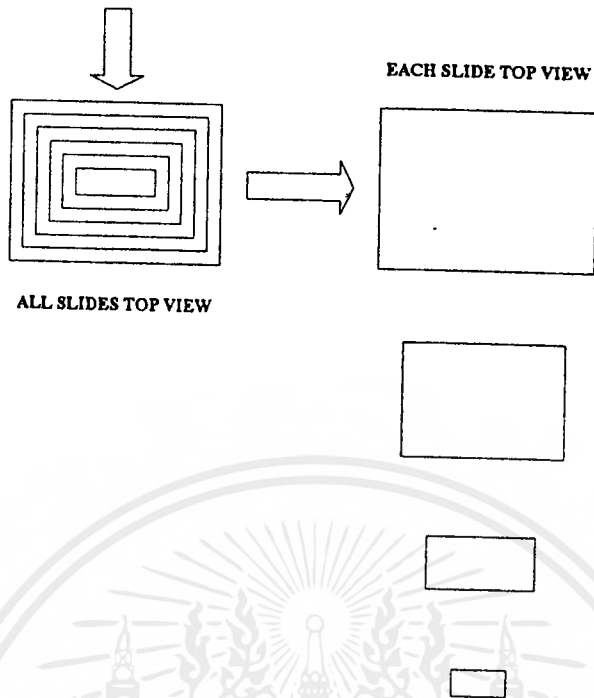
จากรูป 2.1 เป็นรูปทรงปริมาตร ถ้าเราสามารถหั่นปริมาตรออกมาเป็น wafer ได้ เราจะ
ได้ wafer รูปสี่เหลี่ยมที่มีขนาดเล็กลงไปเรื่อยๆ จากล่างขึ้นบน

รูป 2.2 เป็นรูปทรงกระบอก ถ้าเราสามารถหั่นทรงกระบอกออกมาเป็น wafer ได้
เราจะพบว่าได้ wafer ลักษณะวงกลมทุกชิ้น



รูปที่ 2.3 แสดง Top View ของทรงกระบอกและ wafer

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



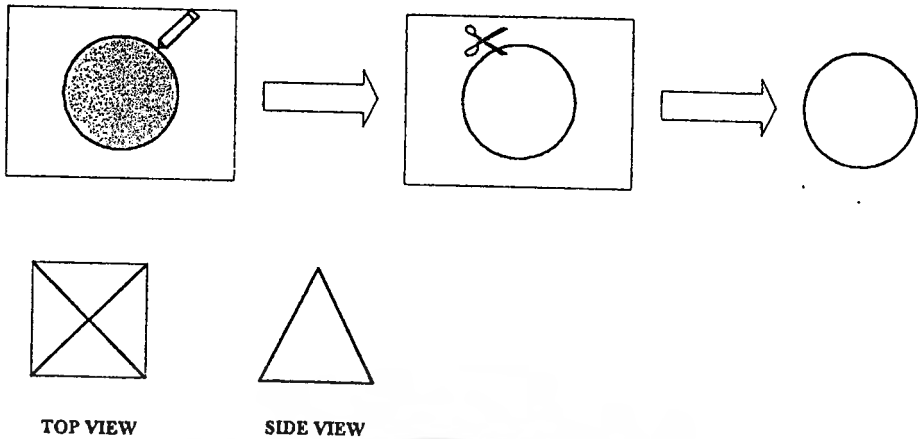
รูปที่ 2.4 แสดง Top View ของปริมาตรและ wafer

ในการสร้างวัตถุกลับมาเป็นรูปทรง 3 มิติ นั้น เราทำโดยการนำ wafer มาวางทับกันไปเรื่อยๆ ก็จะได้ภาพจริงขึ้นมา

2.1 การเก็บข้อมูลของวัตถุเชิง 3 มิติ

จากแนวคิดที่เราจะหันวัตถุออกมาเป็น wafer แล้วเก็บเส้นรอบวงของ wafer นั้นๆ ระบุด้วยว่าเป็น wafer ลำดับที่เท่าไร เราก็จะได้วัตถุเชิง 3 มิติ

การเก็บ wafer ในความเป็นจริง ก็คือ การเก็บเส้นรอบวงของ wafer นั้น ทำได้โดยการวางวัตถุลงบนกระดาษแล้ววาดเส้นรอบวงของ wafer ลงบนกระดาษ แล้วตัดกระดาษ เป็นเสมือนการสร้าง wafer ขึ้นใหม่



รูปที่ 2.5 การสร้าง wafer จากวัตถุดิบ และการนำ wafer กลับมาสร้างเป็นวัตถุดิบ

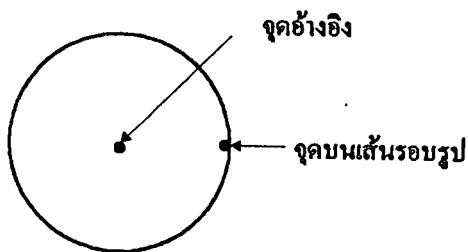
2.1.1 ข้อมูลที่ใช้ในการสร้างภาพ 3 มิติเบื้องต้น

1) ลำดับของ wafer

ลำดับของ wafer เก็บได้ง่ายโดยนับว่า wafer แรกที่นำมาหาเส้นรอบรูปเรียกเป็น wafer ที่ 1 และ wafer ต่อไปเรียกเป็น wafer ที่ 2

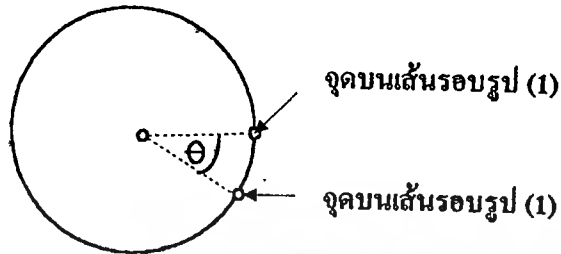
2) เส้นขอบของ wafer

เส้นขอบของ wafer ทำได้โดยหาจุดรอบรูป แล้วทำการเชื่อมต่อจุด ดังนั้นจะสร้างจุดอ้างอิงขึ้นมา 1 จุดบน wafer (จะอยู่ตรงไหนก็ได้) สมมุติว่าอยู่ที่จุดศูนย์กลางของ wafer รูปวงกลม



รูปที่ 2.6 แสดงจุดอ้างอิงของงานหมุนและจุดบนเส้นรอบรูปของวัตถุดิบ

จุดอีกจุดหนึ่งคือจุดที่อยู่บนเส้นรอบรูป หาระยะทางระหว่างจุด 2 จุดนี้
จุดต่อไปหาโดยเปลี่ยนจุดบนเส้นรอบรูปโดยต้องบอกว่าจะมุมเปลี่ยนไปจากจุดแรกเท่าไร



รูปที่ 2.7 แสดงจุด 2 จุดบนเส้นรอบรูปของวัตถุทำมุมระหว่างกัน

ทำไปจนตลอด จำนวนจุดในแต่ละ wafer ขึ้นอยู่กับว่าเราต้องการให้ละเอียดมากเท่าไร ถ้าเรา
ต้องการให้ละเอียดมากก็สร้างจุดบนเส้นรอบรูปให้มากขึ้น



รูปที่ 2.8 แสดงความแตกต่างระหว่างเส้นรอบรูปที่ได้จากการใช้จุด 4 จุด และ 8 จุดต่อชั้น

2.1.2 ข้อมูลที่ใช้ในการสร้างภาพ 3 มิติ

จากการเก็บวิธีนี้พบว่า ข้อมูลที่เราต้องการเก็บจะเพิ่มเป็น 3 พารามิเตอร์ คือ

- 1) ลำดับของ wafer
- 2) ระยะทางจากจุดอ้างอิง
- 3) มุมที่หมุนไปในการเก็บจุดบนเส้นรอบรูปเทียบกับจุดบนเส้นรอบรูปจุดแรก

2.2 การเรียงพารามิเตอร์ทั้ง 3 กลับไปเป็นรูปทรง 3 มิติ

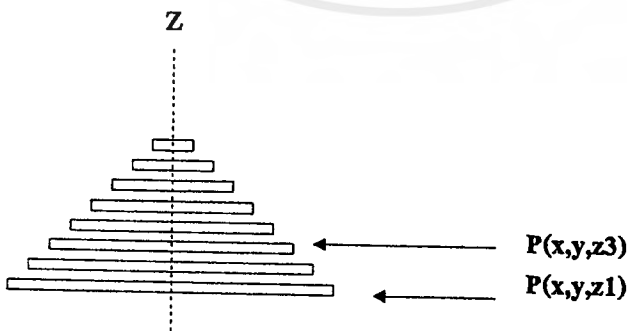
- 1) ต้องวาดจุดบนเส้นรอบวงจุดแรกของ wafer (จุดแรกนี้ให้เป็นจุดที่มีมุมที่หมุนไปเท่ากับ 0 องศา)
- 2) เลื่อนมุมไป ให้ต่างจากจุดแรกเป็นมุมตามพารามิเตอร์ที่เราเก็บไว้ และสร้างจุดที่ 2 ให้ห่างจากจุดอ้างอิงเป็นระยะทางเท่ากับพารามิเตอร์ที่ 2
- 3) วาดจุดต่อไป โดยใช้วิธีตามข้อ 2
- 4) ลากจุดที่มีมุมต่อกันไปเรื่อยๆ จนกลายเป็น wafer



รูปที่ 2.9 แสดงการนำจุดที่ได้มาเรียงกัน โดยแต่ละจุดทำมุมกัน 360 / จำนวนจุดต่อชั้น

2.3 แนวความคิดด้านการเก็บข้อมูลที่จะใช้เป็นอัลกอริธึม

แต่ละ wafer จะต้องมีจุดอ้างอิงอยู่ ถ้าเราให้แต่ละ wafer มีจุดอ้างอิงเดียวกันแต่ต่างกันเพียงความสูง

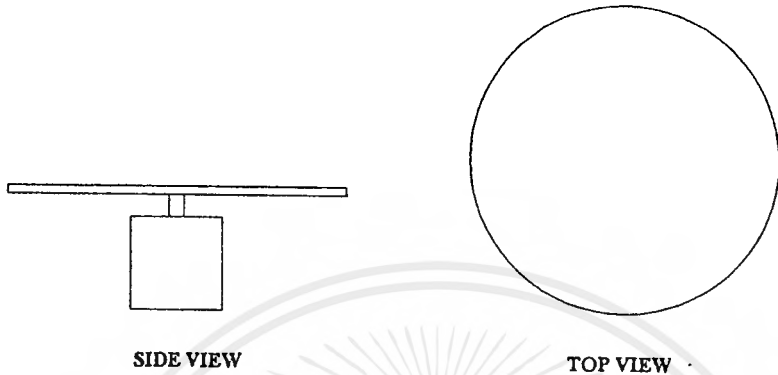


รูปที่ 2.10 แต่ละชั้นมีจุดอ้างอิงเดียวกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซึ่ง x คือ ระยะจากจุดอ้างอิงถึงขอบของ wafer , y คือ จุดอ้างอิง และ z คือ ความสูงของแต่ละชั้น
ดังนั้น ค่า y จะมีค่าเท่ากันทุกชั้นการเก็บข้อมูล

ถ้าเรามีงานรองที่สามารถหมุนได้ (จุดหมุนเป็นจุดศูนย์กลาง)



รูปที่ 2.11 แสดง Top View และ Side View ของงานหมุน

ใช้วิธีการเก็บข้อมูลแบบ 3 พารามิเตอร์

จากที่กล่าวมาในหัวข้อที่แล้ว เราจะทำแบบวัตถุคงที่แล้วหมุนมุมเก็บภาพ ซึ่งถ้าคงที่จุดเก็บภาพและหมุนวัตถุแทน โดยมุมที่ได้แทนที่จะเป็นมุมที่เปลี่ยนไปในการเก็บข้อมูล จะกลายเป็นมุมที่วัตถุหมุนไป วิธีสร้างภาพกลับจะทำเหมือนกับวิธีการหมุนจุดเก็บบนเส้นรอบรูป

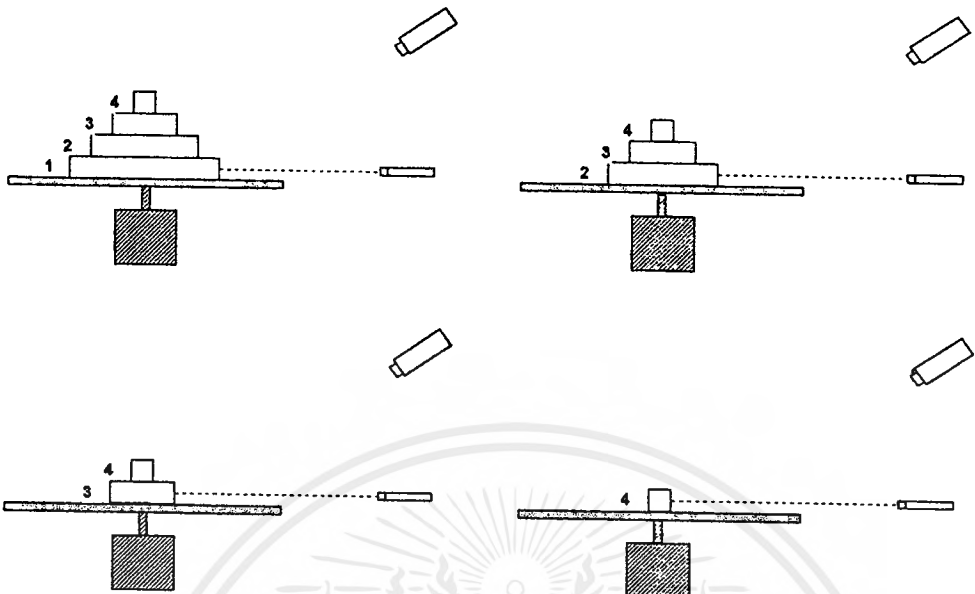
การทำงาน ทำได้โดยเราต้องมี

- กล้องสำหรับถ่ายภาพ แล้วนำไปประมวลผลหาพารามิเตอร์ระยะทาง
- แสงเลเซอร์ ใช้สำหรับยิงไปยังผิววัตถุ ปรากฏเป็นจุดบนเส้นรอบรูปของวัตถุ
- มอเตอร์หมุน งานหมุน ใช้เปลี่ยนมุมในการเก็บภาพ

2.2.1 การหาจุดอ้างอิง

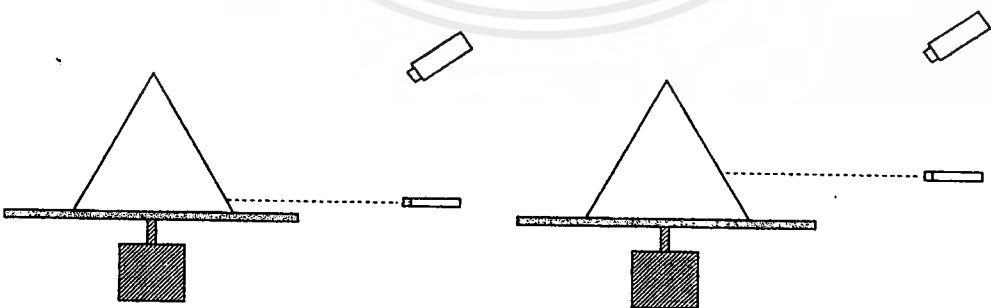
ในการทำงาน เนื่องจากต้องการอ้างอิงที่ y เดียวกันในทุก ๆ ชั้น จะทำได้ง่ายถ้าสามารถหันวัตถุที่นำมาสแกนจริงๆ ได้ แต่ในความเป็นจริงเราไม่สามารถหันวัตถุต้นแบบได้

ถ้าเราหันได้ เราก็สามารถนำชั้นที่สแกนไปแล้วออกไปได้ เราก็จะสามารถขีคกล้องและเลเซอร์ ให้อยู่กับที่ได้ ระยะที่ใช้อ้างอิงเป็นจุดศูนย์กลาง จากรูปที่ถ่ายจากกล้อง จะได้เป็นระยะเดิมนั่นเอง เพราะเราไม่ได้เปลี่ยนมุมกล้องและเลเซอร์



รูปที่ 2.12 ยึดกล้องกับเลเซอร์โดยเอา wafer ออกทีละชั้น

จากแนวคิดการทำงานของวัตถุในกรณีที่สามารถหั่นได้ เปรียบเสมือนเราขีด เลเซอร์และกล้องไว้ด้วยกัน แล้วเปลี่ยน wafer เข้ามาเก็บข้อมูลทีละชั้น แต่เราหั่นวัตถุไม่ได้ จึงยึดวัตถุและเปลี่ยนให้เลเซอร์และกล้องเคลื่อนขึ้นไป เปรียบเสมือนเราเปลี่ยนจานรองหมุนขึ้นไปด้วย คังหัน จุดอ้างอิงจะอยู่ที่เดิม

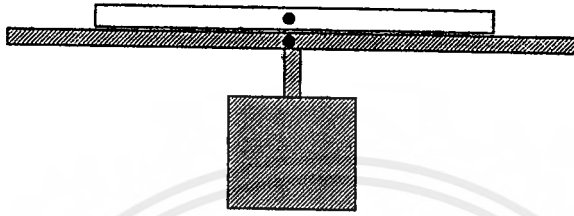


รูปที่ 2.13 เลื่อนกล้องกับเลเซอร์ขึ้นพร้อมกันในการทำ wafer แต่ละชั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การหาระยะจากจุดอ้างอิงทำโดย

พิจารณาการสแกนครั้งแรก wafer ติดอยู่กับงานหมุน และถ้า wafer มีความยาวมาก การที่จะกำหนดจุดอ้างอิงที่ wafer ที่ 1 ก็เปรียบเสมือนกำหนดจุดอ้างอิงที่งานหมุน



รูปที่ 2.14 แสดงจุดอ้างอิงบน wafer และ งานหมุน

บทที่ 3

ทฤษฎีพื้นฐานและหลักการทำงานของสเตปมอเตอร์

3.1 คุณลักษณะทั่วไป

สเตปมอเตอร์ คือ อุปกรณ์ที่เคลื่อนที่เป็นสเตปโดยการกระตุ้นด้วยวิธีการทางแม่เหล็กไฟฟ้า ซึ่งเปลี่ยนสัญญาณดิจิทัลอินพุตซึ่งเป็นพัลส์ไปเป็นการเคลื่อนที่แบบอนาลอกที่เอาท์พุท สเตปมอเตอร์บางครั้งถูกเรียกว่า สเตปปีงมอเตอร์ หรือ สเตปเปอร์มอเตอร์ที่ได้สมญานามเช่นนี้ก็เพราะว่าสเตปมอเตอร์เป็นอุปกรณ์ซึ่งเคลื่อนที่เมื่อถูกกระตุ้นโดยโวลต์เตจหรือกระแส ซึ่งโดยมากจะเป็นไฟฟ้ากระแสตรง เอาท์พุทของสเตปมอเตอร์จะมีจำนวนของคำสั่งอินพุตซึ่งมีลักษณะเป็นพัลส์ โดยเมื่อป้อนกระแสให้กับสเตปมอเตอร์แล้ว สเตปมอเตอร์จะมีการเคลื่อนที่เพิ่มขึ้น 1 สเตป

สเตปมอเตอร์มีมาประมาณ 40 กว่าปีมาแล้วสมัย ก่อนการใช้งานสเตปมอเตอร์มีประสิทธิภาพคำสั่งเอซีมอเตอร์และดีซีมอเตอร์ไม่ได้ แต่เมื่อไม่นานมานี้ได้มีการนำดิจิทัลคอมพิวเตอร์เข้ามาใช้จึงได้เปลี่ยนรูปแบบการควบคุมสเตปมอเตอร์ใหม่ โดยใช้การควบคุมด้วยไมโครโปรเซสเซอร์หรือไมโครคอนโทรลเลอร์ซึ่งจะช่วยให้ใช้งานสเตปมอเตอร์ได้สะดวกขึ้นและมีประโยชน์ในการใช้สอนมากขึ้น ในปัจจุบันนี้สเตปมอเตอร์ได้ถูกนำมาใช้ในอุปกรณ์ที่ใช้ในการควบคุมเชิงเลข (Numerical Control) การควบคุมกระบวนการ (Process Control) และการควบคุมอุปกรณ์ทางเครื่องกล (Machine Tool Control) เป็นต้น

การควบคุมการทำงานของสเตปมอเตอร์โดยทั่วไปใช้วงจรรีบ ซึ่งให้สเตปมอเตอร์ตอบสนองต่อสัญญาณพัลส์ซึ่งทำให้มอเตอร์เคลื่อนที่เป็นจังหวะต่อเนื่องกันไปการเคลื่อนที่ลักษณะเช่นนี้ก่อให้เกิดการอสซิลเลทระหว่างสเตปของการเคลื่อนที่ทำให้สเตปมอเตอร์สั่นระหว่างใช้งาน การแก้ปัญหาดังกล่าวกระทำได้ด้วยวิธีการแดมป์ (Damping Methods) ของสเตปมอเตอร์ ซึ่งวิธีการแก้ปัญหาคตามลักษณะเฉพาะของมอเตอร์แต่ละตัว

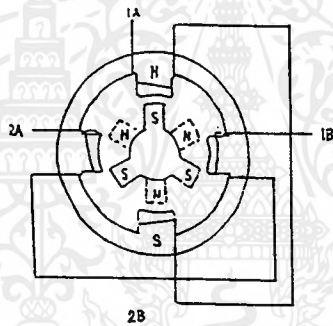
อีกแนวทางหนึ่งที่ใช้ในการแก้ไขปัญหาของการติดต่อรหว่างคอมพิวเตอร์กับมอเตอร์ ก็คือการใช้มอเตอร์ที่ง่าย ต่อการควบคุมโดยคอมพิวเตอร์ สเตปมอเตอร์เป็นมอเตอร์ที่เหมาะสมสำหรับงานเช่นนี้มากเพราะ มันสามารถควบคุมให้หมุนได้โดยใช้สถานะเพียง 2 สถานะ คือ ON และ OFF ซึ่งสัญญาณที่เข้าเครื่องคอมพิวเตอร์สามารถสร้างขึ้นอย่างง่ายโดย

สเตปมอเตอร์มีขดลวดอยู่หลายขด โดยที่แต่ละขดถูกควบคุมการจ่ายกระแสโดยสัญญาณ ON-OFF ทำให้สามารถควบคุมตำแหน่งในการเคลื่อนที่ได้ครบโดที่ขั้ว ไม่เกิดการเกินกำลัง (Overload) คราวนั้นมันยังคงรักษาตำแหน่งเดิม จนกระทั่งมีการกระตุ้นโดยสัญญาณควบคุมการ

จ่ายกระแสอีกครั้ง และเนื่องจากสเตปมอเตอร์สามารถควบคุมตำแหน่งได้โดยการส่งสัญญาณ กระตุ้น ทำให้ไม่จำเป็นต้องใช้ตัววัดตำแหน่ง (Encoder) ในการทำงาน ข้อเสียที่สำคัญของ สเตปมอเตอร์ก็คือ มีข้อจำกัดด้านความเร็วเมื่อเปรียบเทียบกับ DC Servo Motor ที่มีขนาดเท่ากัน และยังมีการสั่นสะเทือน (Vibration) สูงทำให้การเคลื่อนที่เป็นไปอย่างไม่ราบเรียบ

3.2 หลักการทำงานของสเตปมอเตอร์

สเตปมอเตอร์สามารถแบ่งโครงสร้างทางกายภาพได้เป็น 2 ส่วน คือ สเตเตอร์ (Stator) และ โรเตอร์ (Rotor) ตัวสเตเตอร์เป็นส่วนที่อยู่กับที่ ประกอบด้วยขดลวดทองแดงซึ่ง พันอยู่รอบแกนเหล็ก เพื่อสร้างสนามแม่เหล็กเมื่อมีการจ่ายกระแสผ่านขดลวด ส่วนโรเตอร์เป็น ส่วนที่เคลื่อนที่มีลักษณะเป็นแท่งเหล็กทรงกลม และที่ผิวรอบนอกมีลักษณะเป็นซิกฟันซึ่งทำจาก แม่เหล็กถาวร ดังรูป 3.1

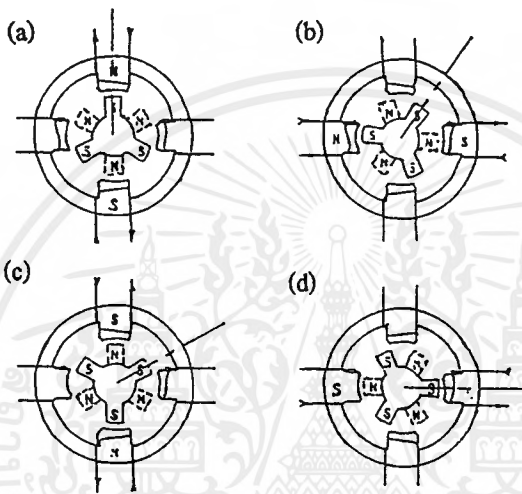


รูป 3.1 แสดงโครงสร้างของไฮบริดจ์สเตปมอเตอร์ที่มีจำนวนสเตปต่อรอบเท่ากับ 12

เมื่อยังไม่มีการจ่ายกระแสให้กับขดลวดของมอเตอร์ ซิกฟันอันใดอันหนึ่งของโรเตอร์จะ อยู่ในตำแหน่งที่ตรงกันกับซิกฟันอันใดอันหนึ่งของสเตเตอร์ ทั้งนี้เป็นเพราะแม่เหล็กถาวรที่ตัวของ โรเตอร์พยายามที่จะทำให้ค่าความต้านทานทางแม่เหล็กไฟฟ้า (Reluctance) มีค่าน้อยที่สุด ซึ่งณ.ที่จุดที่ซิกฟันของตัวโรเตอร์และสเตเตอร์ตรงกันนั้นมีค่าความต้านทานทางแม่เหล็กไฟฟ้าน้อย ที่สุดทำให้เกิดเส้นแรงแม่เหล็กไฟฟ้ามากที่สุด และจากรูปที่ 3.1 เส้นแรงแม่เหล็กไฟฟ้าจะทำให้ เกิดขั้วแม่เหล็กเหนือและใต้ขึ้นมา 2 คู่ ทั้งที่ตัวสเตเตอร์และตัวโรเตอร์ดังรูป ค่าทอร์ก (Torque) ที่ทำให้ตัวโรเตอร์สามารถยึดอยู่ในตำแหน่งดังกล่าวนี้เรียกว่า ดีเท็นทอร์ก (Detent Torque) (หมายความว่า การที่จะทำให้มอเตอร์เคลื่อนที่ในขณะที่ไม่ได้จ่ายกระแสให้กับขดลวดของมอเตอร์

จะต้องออกแรงมากกว่าค่าของคิเห็นทอร์ค จึงจะทำให้โรเตอร์เริ่มเคลื่อนที่ได้) รูป 3.1 นั้นมีตำแหน่งที่สามารถเกิดคิเห็นทอร์คได้

เมื่อจ่ายกระแสให้กับขดลวดที่อยู่ในสเตเตอร์คู่ใดคู่หนึ่ง ดังรูป 3.2(a) จะทำให้เกิดขั้วแม่เหล็กเหนือและใต้ที่ซีกฟันของตัวสเตเตอร์ ซึ่งจะดึงดูดซีกฟันของตัวโรเตอร์ที่มีขั้วแม่เหล็กที่มีศักย์ต่างกันที่จำนวนตำแหน่งเท่ากับจำนวนซีกฟันของโรเตอร์ และแรงที่จะทำให้โรเตอร์เปลี่ยนตำแหน่งไปจากตำแหน่งสเตเบิลโพสิชันได้นี้เรียกว่า โฮลดิ้ง ทอร์ค (Holding Torque)

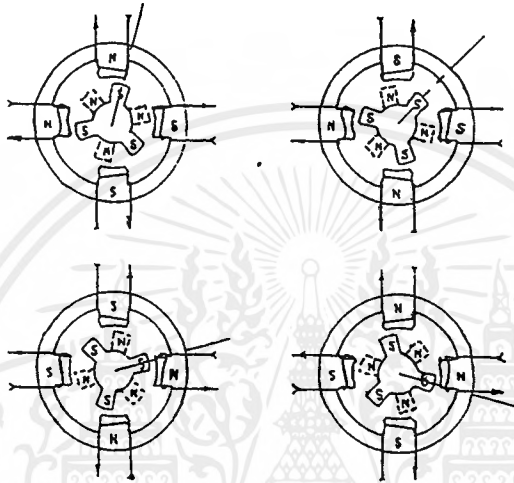


รูป 3.2 แสดงขั้นตอนการทำงานของสเตปมอเตอร์แบบเต็มสเตปหนึ่งเฟส

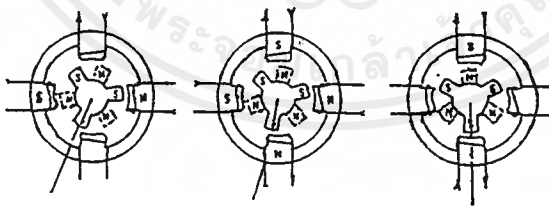
เมื่อสลับเปลี่ยนการจ่ายกระแสให้แก่ขดลวด จากขดหนึ่งไปอีกขดหนึ่งเนื่องจากขดลวดวางอยู่ในตำแหน่งที่ต่างกัน 90 องศาจะทำให้ตัวสเตเตอร์ดึงดูดซีกฟันของตัวโรเตอร์อีกซีกหนึ่ง ที่ใกล้ที่สุดเข้าไว้ ซึ่งจะทำให้โรเตอร์เคลื่อนที่ไป 1 สเตปหรือ 30 องศา ดังรูปที่ 3.2(b) จากนั้นก็เปลี่ยนไปจ่ายกระแสให้กับขดลวดชุดแรกโดยในคราวนี้เปลี่ยนทิศทางกระแสให้ตรงข้ามกับครั้งแรก ซึ่งจะทำให้ตัวโรเตอร์เคลื่อนที่ไปอีก 1 สเตป (เคลื่อนที่ไป 30 องศา) ดังรูปที่ 3.2 (c) หลังจากนั้นก็ไปจ่ายกระแสให้กับขดลวดชุดที่สองโดยกลับทิศทางของกระแสที่ป้อนให้อีกเช่นกัน ทำให้โรเตอร์หมุนไป 90 องศา ดังรูป 3.2(d) และถ้าหากเราป้อนกระแสให้กับมอเตอร์เหมือนที่เราป้อนในครั้งแรกแล้ว ซีกฟันซีกถัดไปของตัวโรเตอร์จะอยู่ในตำแหน่งที่เหมือนกับในรูปที่ 3.2(a) อีกครั้งหนึ่ง ถ้าหากเราต้องการเคลื่อนที่หนึ่งรอบ เราต้องทำการกระตุ้นให้มอเตอร์เคลื่อนที่ไปจนครบ 12 สเตป และถ้าต้องการให้โรเตอร์หมุนไปอีกทิศทางหนึ่ง

ก็จะทำการสลับลำดับในการจ่ายกระแส จากรูปที่ 3.2(a), (b), (c) และ (d) ไปเป็น (a), (d), (c) และ (b) ตามลำดับ

ถ้าจ่ายกระแสให้แก่ขดลวดทั้งสองขดพร้อมๆ กัน ซีคพื้นของโรเตอร์จะอยู่ ณ ตำแหน่งระหว่างซีคพื้นของสเตเตอร์ เพราะฉะนั้นการจ่ายกระแสให้แก่มอเตอร์แบบนี้จะให้ทอร์คมากกว่าแบบที่จ่ายกระแสในเวลาใดขณะหนึ่งเพียงขดเดียว ดังรูปที่ 3.3 ซึ่งแสดงถึงการป้อนกระแสให้กับขดลวดแต่ละขดเพื่อให้สเตปมอเตอร์เคลื่อนที่



รูปที่ 3.3 แสดงขั้นตอนการทำงานของสเตปมอเตอร์แบบเต็มสเตปสองเฟส



รูปที่ 3.4 แสดงขั้นตอนการทำงานของสเตปมอเตอร์แบบครึ่งสเตป

แต่ถ้าในตอนแรกจ่ายกระแสให้กับขดลวดพร้อมๆ กันสองขด การทำอย่างนี้สลับกันไปจะทำให้โรเตอร์เคลื่อนที่ไปสเตปละ 15 องศา ดังรูปที่ 3.4 และในการขับเคลื่อนแบบนี้จะทำให้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สเตปต์อรอบ (Step/Rep) เพิ่มขึ้นเป็นเท่าตัว การจับสเตปมอเตอร์แบบนี้เรียกว่า การจับแบบครึ่งสเตป (Half Stepping) ซึ่งนิยมใช้กันในงานอุตสาหกรรมเป็นอย่างมาก แม้ว่าในบางครั้งจะให้ทอร์คน้อยกว่า แต่มีข้อดีว่าการจับแบบเต็มสเตป (Full Stepping) ก็คือ การเคลื่อนที่เป็นไปอย่างราบเรียบ (ที่ความเร็วไม่สูงนัก) อีกทั้งมีความแม่นยำสูงและการสั่นสะเทือนน้อยกว่า

3.3 การ Control Step Motor

วิธีการควบคุมการทำงานของสเตปมอเตอร์โดยการกระตุ้นเฟสมี 3 วิธี คือ

3.3.1 การกระตุ้นแบบซิงเกิลเฟส (Single Phase Excitation)

เป็นการกระตุ้นแบบเฟสเดียวตามสัญญาณพัลส์ที่ป้อนเข้ามาที่จุดจับสเตปมอเตอร์ ดังนี้

PHASE	PULSE							
A	1	0	0	0	1	0	0	0
B	0	1	0	0	0	1	0	0
C	0	0	1	0	0	0	1	0
D	0	0	0	1	0	0	0	1

3.3.2 การกระตุ้นแบบสองเฟส (Two Phase Excitation)

เป็นการกระตุ้นแบบทีละสองเฟสคู่พร้อมกัน ดังนี้

PHASE	PULSE							
A	1	0	0	1	1	0	0	1
B	1	1	0	0	1	1	0	0
C	0	1	1	0	0	1	1	0
D	0	0	1	1	0	0	1	1

3.3.3 การกระตุ้นแบบครึ่งสเตป (Half Step Excitation)

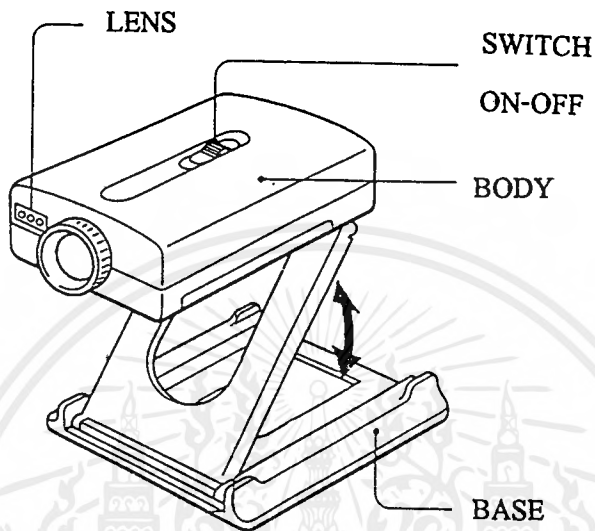
เป็นการรวมรูปแบบการหมุนของทั้งสองแบบไว้ในแบบเดียวกัน

PHASE	PULSE								
A	1	1	0	0	0	0	0	0	1
B	0	1	1	1	0	0	0	0	0
C	0	0	0	1	1	1	0	0	0
D	0	0	0	0	0	1	1	1	1

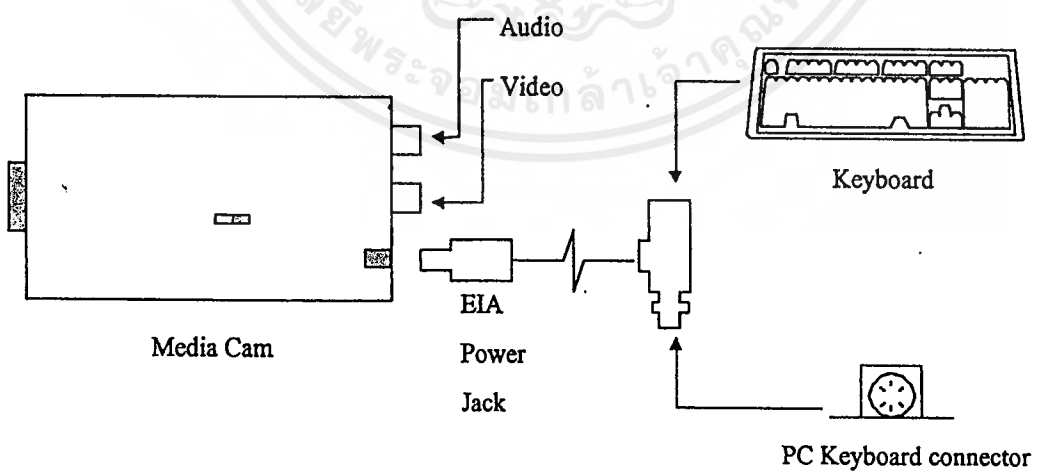
วิธีการกระตุ้นเฟสทั้ง 3 แบบนี้จะมีคุณสมบัติที่แตกต่างกัน ดังนั้นการเลือกวิธีการกระตุ้นจึงจำเป็นที่จะต้องพิจารณาเพื่อให้เหมาะสมกับงานนั้นๆ การกระตุ้นแบบเฟสเดียว จะเป็นแบบที่มีความเที่ยงตรงของตำแหน่งสูง แต่จะมีแรงบิด (Torque) น้อย ส่วนการกระตุ้นแบบ 2 เฟส มีความเที่ยงตรงของตำแหน่งน้อยกว่าแบบแรก แต่มีแรงบิดสูงกว่า และการกระตุ้นแบบครึ่งสเตปเป็นแบบที่มีความเที่ยงตรงของตำแหน่งน้อยมาก แต่จะมีแรงบิดสูงมากเช่นกัน

บทที่ 4

กล้อง Color Video Camera



รูปที่ 4.1 แสดงลักษณะของกล้อง Color Video Camera



รูปที่ 4.2 แสดงการต่อกล้องเข้ากับเครื่อง PC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1 สิ่งที่เป็นต้องมีในการใช้งานกล้อง Color Video Camera ได้แก่

- เครื่อง PC Pentium 75 ขึ้นไป
- Windows 95
- RAM 8 MB
- PCI 1 slot
- พื้นที่ Harddisk 10 MB
- Card จอภาพแบบ SVGA

4.2 คุณสมบัติพิเศษของกล้อง

- ให้ภาพที่มีความละเอียด 640 x 480 pixel 24 bit color
- Frame Rate 30 fps
- ได้รับไฟเลี้ยงจากคีย์บอร์ดด้วยกำลังไฟอย่างน้อย 2 W
- ปรับไฟกัศได้จาก 1 นิ้ว ถึงอินฟินิตี้



บทที่ 5

คณิตศาสตร์ในการสร้างโครงสร้าง 3 มิติ

5.1 Two-Dimensional Coordinate Geometry

ในงานด้าน Computer Graphics นั้น รูปร่างและขนาดของวัตถุที่แสดงเป็น 2 มิติจะต้องใช้ระบบ coordinate ซึ่งปกติจะใช้ x, y ในระบบ Cartesian Coordinate ดังนั้นการ Transform รูปแบบต่างๆในทางเรขาคณิต จึงได้ถูกนำไปใช้เป็น Model สำหรับการ Shift , Resize หรือ Reorient Coordinate เหล่านั้น

พื้นฐานของ Model 2 มิติ คือ จุด ยกตัวอย่างเช่น เส้นก็คือจุด 2 จุดต่อกัน หรือ ระนาบจะประกอบไปด้วยจุดหลายๆจุดที่อยู่บนระนาบ ดังนั้นในระบบ 2 มิติจะถูกกำหนดขึ้นด้วยกลุ่มของ Coordinate x, y หรือจุดใดๆนั่นเอง

เราจะทำการสมมติรูป 3 เหลี่ยมในระนาบ 2 มิติซึ่งสามารถเขียนเป็น Matrix ขนาด 3×2 ได้ดังนี้

$$[P] = \begin{bmatrix} x_1 & y_1 \\ x_2 & y_2 \\ x_3 & y_3 \end{bmatrix} \quad (5.1)$$

ซึ่ง x และ y นี้เป็นเวกเตอร์ตำแหน่งที่มีความสัมพันธ์กับระบบ Coordinate แบบ Cartesian ถ้าเราใช้ระบบ Coordinate แบบ Cartesian แล้วเราจะพบกับข้อจำกัดในการ Transform ซึ่งในการ Transform เราจะต้องเกี่ยวข้องกับ Operation การคูณ ซึ่งมีข้อจำกัดในเรื่องขนาดของ Matrix และเงื่อนไขของเวกเตอร์ ดังนั้นเราจึงหลีกเลี่ยงปัญหาเหล่านี้ด้วยการ Transform โดยใช้ระบบ Coordinate แบบ Homogeneous เพื่อความเข้าใจระบบ Coordinate แบบ Homogeneous แล้วเราจะสมมติจุด $p(x, y)$ กับจุด origin จะถูกแสดงโดยพารามิเตอร์ h ดังนั้นเราสามารถเขียนสมการได้คือ

$$P(x, y, z) = P(hx, hy, h) \quad (5.2)$$

ซึ่งเราจะมองจุด $p(x, y)$ นี้ในลักษณะของระนาบ 3 มิติโดย h ก็จะขึ้นค่าของ z นั่นเอง เพราะฉะนั้นจุด 2 มิติใดๆก็สามารถแสดงเป็นจุดในระนาบ 3 มิติได้ซึ่งเราเรียกระนาบเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นี่ว่า Homogenous แต่ยกเว้นในกรณีที่จุดนั้นเป็นจุด origin (ซึ่งค่า $h=0$) ตัวอย่างเช่นเมื่อมีจุด $P(2,4)$ ใน Cartesian Coordinate จะทำให้เกิดจุด $P(4, 8, 2), P(6, 12, 3), P(2, 4, 1), P(m, n, h)$ ใน Homogeneous Coordinate ดังนั้นถ้าเรามีจุดต่างๆเหล่านี้จะทำให้เราสามารถหาจุดใน Cartesian Coordinate ได้คือ $P(m/h, n/h, 1)$

$$\begin{aligned} \text{โดยที่ } x &= m/h \\ y &= n/h \end{aligned} \quad (5.3)$$

เมื่อใช้ Homogeneous Coordinate แล้วค่าต่างๆในระนาบ 2 มิติ จะถูกแสดงในรูปของ Matrix ขนาด $[n \times 3]$ ซึ่ง $n =$ จำนวนของจุดของวัตถุ ดังนั้น

$$[P] = \begin{bmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{bmatrix} \quad (5.4)$$

5.1.1 การ Scaling 2 มิติ

การทำ Scaling คือ การทำให้วัตถุมีขนาดที่เพิ่มขึ้น การ Scaling ในทิศทาง x และ y ด้วยค่าคงที่นั้นจะเกิดการเปลี่ยนแปลงของความยาวเกิดขึ้น ถ้ามีขนาดมากกว่า 1 จะเป็นการขยาย แต่ถ้าน้อยกว่า 1 จะเป็นการลดขนาด แต่ไม่ว่าจะเป็นกรณีใดจะต้องเป็นค่าบวก เพราะถ้าเป็นค่าลบแล้วจะเกิดการ Reflection ขึ้น

การ Scaling จุด $P(x, y)$ เป็น $P(x^*, y^*)$ แสดงได้ดังสมการ (5.5)

$$\begin{aligned} x^* &= x(S_x) \\ y^* &= y(S_y) \end{aligned} \quad (5.5)$$

หรือแสดงในรูป Matrix ได้ดังสมการ (5.6)

$$[x^* \ y^* \ 1] = [x \ y \ 1] \begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (5.6)$$

ในการ Scaling นี้เป็นการ Scaling ที่ใช้จุด origin เป็นจุดอ้างอิง ซึ่งแต่ละจุดบน วัตถุนั้นจะเปลี่ยนขนาดหรือทิศทางก็ตามต้องขึ้นอยู่กับจุด origin เสมอ

5.1.2 การ Translation 2 มิติ

การ Translation นั้นจะทำให้วัตถุเปลี่ยนตำแหน่งไปในทิศทางตามค่าที่กำหนดไป ซึ่งสามารถแสดงให้เห็นตามสมการ (5.7)

$$\begin{aligned}x^* &= x + T_x \\y^* &= y + T_y\end{aligned}\quad (5.7)$$

แสดงในรูป Matrix ตามสมการ (5.8)

$$[x^* \ y^* \ 1] = [x \ y \ 1] \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ T_x & T_y & 1 \end{bmatrix}\quad (5.8)$$

5.1.3 การ Rotation 2 มิติ

การ Rotation นั้นจะใช้สำหรับการมองวัตถุในทิศทางต่างๆกัน ในการ Rotation จะต้องใช้ origin เป็นจุดอ้างอิงเช่นกัน ตามค่าของมุม θ ที่กำหนดไป ในการ Rotate จุด $P(x, y)$ เป็น $P(x^*, y^*)$ ได้ดังสมการ (5.9)

$$\begin{aligned}x &= r \cos \theta \\y &= r \sin \alpha\end{aligned}\quad (5.9)$$

ซึ่งมุม α เป็นค่าพารามิเตอร์ที่แทนค่ามุมที่กวาดไปในทิศทางตามแกน x

$$\begin{aligned}x^* &= r \cos(\alpha + \theta) = r \cos \alpha \cos \theta - r \sin \alpha \sin \theta \\y^* &= r \sin(\alpha + \theta) = r \sin \alpha \cos \theta + r \cos \alpha \sin \theta\end{aligned}\quad (5.10)$$

เมื่อแทนค่า x, y จากสมการ (5.9) จะได้สมการดังนี้

$$\begin{aligned}x^* &= x \cos \theta - y \sin \theta \\y^* &= x \sin \theta + y \cos \theta\end{aligned}\quad (5.11)$$

หรือแสดงในรูป Matrix ได้ดังสมการ (5.12)

$$[x^* \ y^* \ 1] = [x \ y \ 1] \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (5.12)$$

5.2 Three - Dimentional Coordinate Geometry

ในการ Design งานทางด้านวิศวกรรมส่วนมากจะเกี่ยวข้องกับวัตถุที่เป็น 3 มิติโดยปกติรูปทรงหรือรูปแบบจำนวนมากจะถูกใช้งานในการสร้างรูปทรง 3 มิติ ซึ่งสามารถแสดงออกมาบนหน้าจอ Display เช่น ผิว Poligon Surface หรือ Framework ของเส้น สำหรับการใช้งานด้านอื่นๆ เช่น การออกแบบเครื่องจักรกล หรือ Body ของเครื่องบิน เป็นการออกแบบที่พิเศษซึ่งถูกใช้เพื่อพัฒนารูปแบบของพื้นผิว 3 มิติ จนกระทั่งได้เงื่อนงำที่พอใจ

เมื่อใดที่วิธีการเหล่านี้ถูกใช้งาน วัตถุ 3 มิติก็จะถูกสร้างขึ้นในระบบ Coordinate 3 มิติ และจะทำการ Map ไปในระบบ 2 มิติเพื่อที่จะทำการ Display การสร้างภาพของวัตถุจำเป็นต้องใช้การ Transform Coordinate 3 มิติ ซึ่งแสดงได้โดย Matrix 4x4 ดังนี้

$$[P] = \begin{bmatrix} 0 & 0 & 0 & 1 \\ x_B & 0 & 0 & 1 \\ 0 & y_C & 0 & 1 \\ 0 & 0 & z_D & 1 \end{bmatrix} \quad (5.13)$$

5.2.1 การ Transformation 3 มิติ

การ Transform ในระบบ 3 มิติ จะถูกใช้ด้วยวิธีการเดียวกับที่ใช้ในระบบ 2 มิติดังที่อธิบายในส่วนการ Transformation 2 มิติ ข้างต้น แต่จะเพิ่ม Coordinate ของ z เข้ามาด้วยซึ่งในระบบ 2 มิติจะประกอบไปด้วย Coordinate x และ y เท่านั้น

ในระบบ Homogeneous Coordinate จุดต่างๆในระนาบ 3 มิติก็จะถูกแสดงเป็น Matrix ขนาด 4x4 ดังนี้

$$\begin{bmatrix} a & b & c & 0 \\ d & e & f & 0 \\ g & h & i & 0 \\ j & k & l & s \end{bmatrix} \quad (5.14)$$

5.2.2 การ Scaling 3 มิติ

ในการทำ Scaling Transform จะทำได้โดยการใส่ค่าไปในแนวทแยงมุมของ Matrix ที่ใช้ในการ Transform ขนาด 4×4 ดังเช่นการ Scaling จุด $P(x, y, z, 1)$ ไปเป็น $P(x^*, y^*, z^*, 1)$ ได้สมการดังนี้

$$[x^*, y^*, z^*, 1] = [x, y, z, 1] \begin{bmatrix} a & 0 & 0 & 0 \\ 0 & e & 0 & 0 \\ 0 & 0 & i & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.15)$$

ด้วยวิธีการนี้เราจะทำการ Scaling โดยใช้จุด origin เป็นจุดอ้างอิงและใช้หลักการเช่นเดียวกับการทำ Scaling ใน 2 มิติ ถ้าตัวแปรในการทำ Scaling คือ a, e, i มีค่าที่ไม่เท่ากันแล้วทำให้ภาพและขนาดของวัตถุที่ได้จะผิดเพี้ยนไป แต่ ถึงแม้จะเกิดการเปลี่ยนแปลงขนาดเกิดขึ้นแต่จุด origin ยังเหมือนเดิม

$$\begin{aligned} [x^*, y^*, z^*, 1] &= [x, y, z, 1] \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & s \end{bmatrix} \\ &= [x, y, z, s] \end{aligned} \quad (5.16)$$

จากวิธีการดังกล่าว เมื่อเราใช้การ Scaling วิธีการนี้แล้วจะสังเกตเห็นว่าใน column ที่ 4 ของ Matrix ตรงจุดที่ใช้ในการ Transform อาจจะไม่เท่ากับ 1 ก็ได้ ดังเช่นหลักการของ 2 มิติที่กล่าวแล้วข้างต้น ซึ่ง Matrix ที่ใช้นี้อาจถูกเปลี่ยนแปลงได้เพื่อให้ค่าของ x, y, z นี้เป็นค่า Coordinate ที่เป็นมาตรฐาน ซึ่งก็คือระบบ Cartesian Coordinate ดังนั้นสมการ (5.16) จะเขียนได้ดังนี้

$$[x, y, z, s] = [x/s, y/s, z/s, 1] \quad (5.17)$$

เมื่อค่า s มีค่ามากกว่า 1 จะเป็นการลดขนาดของวัตถุลง แต่เมื่อต้องการที่ขยายขนาดของวัตถุต้องใช้ค่า $1/s$ ใน Matrix ที่ทำ Scaling ซึ่งก็จะได้ค่าของ Coordinate สุดท้ายเป็น

$$[sx, sy, sz, 1] \quad (5.18)$$

5.2.3 การ Translation 3 มิติ

จากสมการ (5.18) Matrix ที่ใช้ในการ Transform นี้จะทำการ Translate หรือ เคลื่อนย้ายจุด (x, y, z) เป็นจุดใหม่ (x^*, y^*, z^*) โดยใช้ (j, k, l)

$$[x^*, y^*, z^*, 1] = [x, y, z, 1] \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ j & k & l & 1 \end{bmatrix} \quad (5.19)$$

ค่าของ j, k, l นี้ จะแสดงความสัมพันธ์ของการเคลื่อนย้ายจุดไปตามทิศทางของ x, y, z ยกตัวอย่างเช่นเราจะทำการ Translate รูปลูกบาศก์ขนาด 1 หน่วยซึ่งมีจุดยอดจุดหนึ่งอยู่ที่ตำแหน่ง origin ไปที่ตำแหน่งใหม่ในระนาบที่เราพิจารณา โดยใช้ Matrix ที่ใช้ Matrix ที่ใช้สำหรับการทำ Translation ดังสมการ (5.19) แต่ละจุดยอดจะถูก Translate ไปตามค่าที่กำหนดใน Matrix ดังนั้นค่า Coordinate ที่ถูก Transform ของลูกบาศก์นี้จะมีค่าดังสมการ (5.20)

$$[P^*]_{\text{CUBE}} = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ J & K & L & 1 \end{bmatrix} = \begin{bmatrix} 2 & 2 & 0 & 1 \\ 3 & 2 & 0 & 1 \\ 3 & 2 & 1 & 1 \\ 3 & 2 & 1 & 1 \\ 2 & 3 & 1 & 1 \\ 3 & 3 & 1 & 1 \\ 3 & 3 & 0 & 1 \\ 2 & 3 & 0 & 1 \end{bmatrix} \quad (5.20)$$

5.2.4 การ Rotation 3 มิติ

การ Rotation ในระบบ 3 มิติเป็นเรื่องที่สำคัญสำหรับการทำความเข้าใจรูปร่างของ วัตถุ หรือการพิจารณางานที่ถูกออกแบบมาในมุมที่แตกต่างกัน ซึ่งเป็นสิ่งที่ซับซ้อนมากกว่าการ Rotation ในระบบ 2 มิติ เพราะในระบบ 3 มิติต้องระบุแกนในการทำ Rotation ด้วยในขณะที่

ที่ระบบ 2 มิติใช้เพียงจุดในการทำ Rotation ตามแกนใดๆก็ตาม เราสามารถที่จะคิดอย่างง่ายๆ โดยคิดแยกเป็นการหมุนตามแกนหลักทั้ง 3 แกนคือ แกน x, y และ z แล้วนำมาสร้างตามวิธีการเดียวกันกับที่ใช้ในการ Rotation 2 มิติ ทำให้เราได้ Matrix สำหรับการ Rotation สำหรับการ Rotation ตามแกน z ดังสมการ (5.21)

$$[T_R]_z = \begin{bmatrix} \cos\theta & \sin\theta & 0 & 0 \\ -\sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.21)$$

ซึ่งจะได้ผลลัพธ์ของการ Mapping ดังสมการ (5.22)

$$\begin{aligned} x^* &= x \cos\theta - y \sin\theta \\ y^* &= x \sin\theta + y \cos\theta \\ z^* &= z \end{aligned} \quad (5.22)$$

ในวิธีการเดียวกันเมื่อเราทำการ Rotate เป็นมุม θ ไปตามแกน y จะได้สมการ (5.23)

$$[T_R]_y = \begin{bmatrix} \cos\theta & 0 & -\sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ \sin\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.23)$$

ซึ่งจะได้ผลลัพธ์ของการ Mapping ดังสมการ (5.24)

$$\begin{aligned} x^* &= x \cos\theta + z \sin\theta \\ y^* &= y \\ z^* &= -x \sin\theta + z \cos\theta \end{aligned} \quad (5.24)$$

เมื่อ Rotate ไปตามแกน x จะได้ผลดังสมการ (5.25)

$$[T_R]_x = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \theta & \sin \theta & 0 \\ 0 & -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.25)$$

ซึ่งจะได้ผลลัพธ์ของการ Mapping ดังสมการ (5.26)

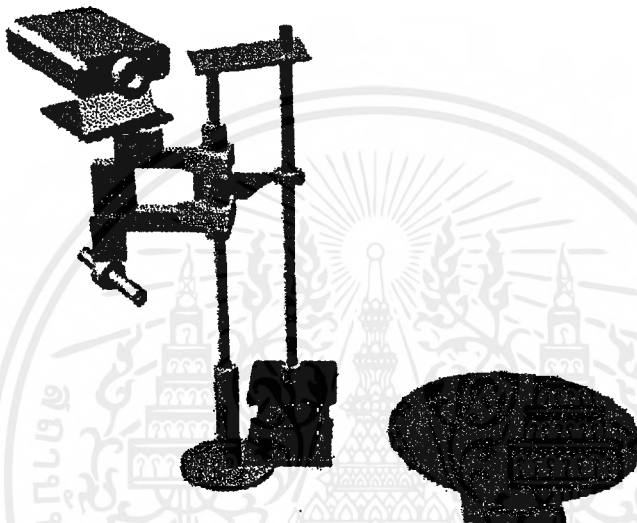
$$\begin{aligned} x^* &= x \\ y^* &= y \cos \theta - z \sin \theta \\ z^* &= y \sin \theta + z \cos \theta \end{aligned} \quad (5.26)$$

ข้อสังเกตที่สำคัญคือเครื่องหมายในเทอม sine ใน Matrix $[T_R]_y$ ซึ่งจะมีค่าตรงกันข้ามกับ Matrix $[T_R]_z$, $[T_R]_x$ เนื่องจากผลลัพธ์ของการใช้กฎมือขวา

บทที่ 6

ชุดทดลองต้นแบบ

6.1 ส่วนประกอบของชุดทดลอง



รูป 6.1 ชุดทดลองต้นแบบ

6.1.1 กล้อง Color Video Camera

สำหรับจับภาพเพื่อนำไปประมวลผล

6.1.2 ชุดกำเนิดเลเซอร์

ชุดกำเนิดเลเซอร์นี้จะให้ลำแสงเลเซอร์ขนาดเล็กเพื่อให้เกิดจุดตัดของแสงกับวัตถุ

6.1.3 Step Motor

มี 2 ตัว ทำหน้าที่ยกกล้องขึ้นลงและหมุนงานที่วางวัตถุแต่ละ step

6.1.4 Barring ยกกล้อง

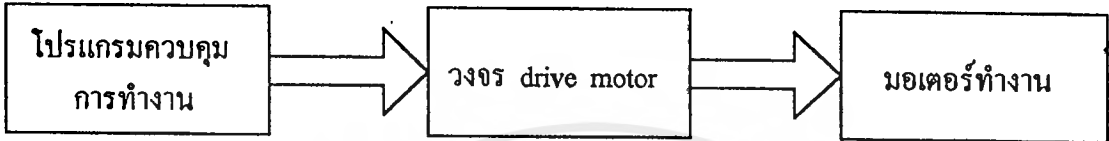
ประกอบด้วยแท่งเหล็กเก็ยขั้วนำและ Barring ที่ลดความเสียดทานโดยมีลูกปืนอยู่ภายใน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.1.5 ชุดจ่ายไฟและ Drive Motor

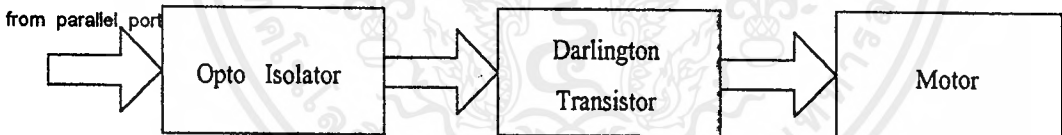
ใช้ไฟ DC ขนาด 5 V ในการขับ Motor

6.2 การขับ Motor

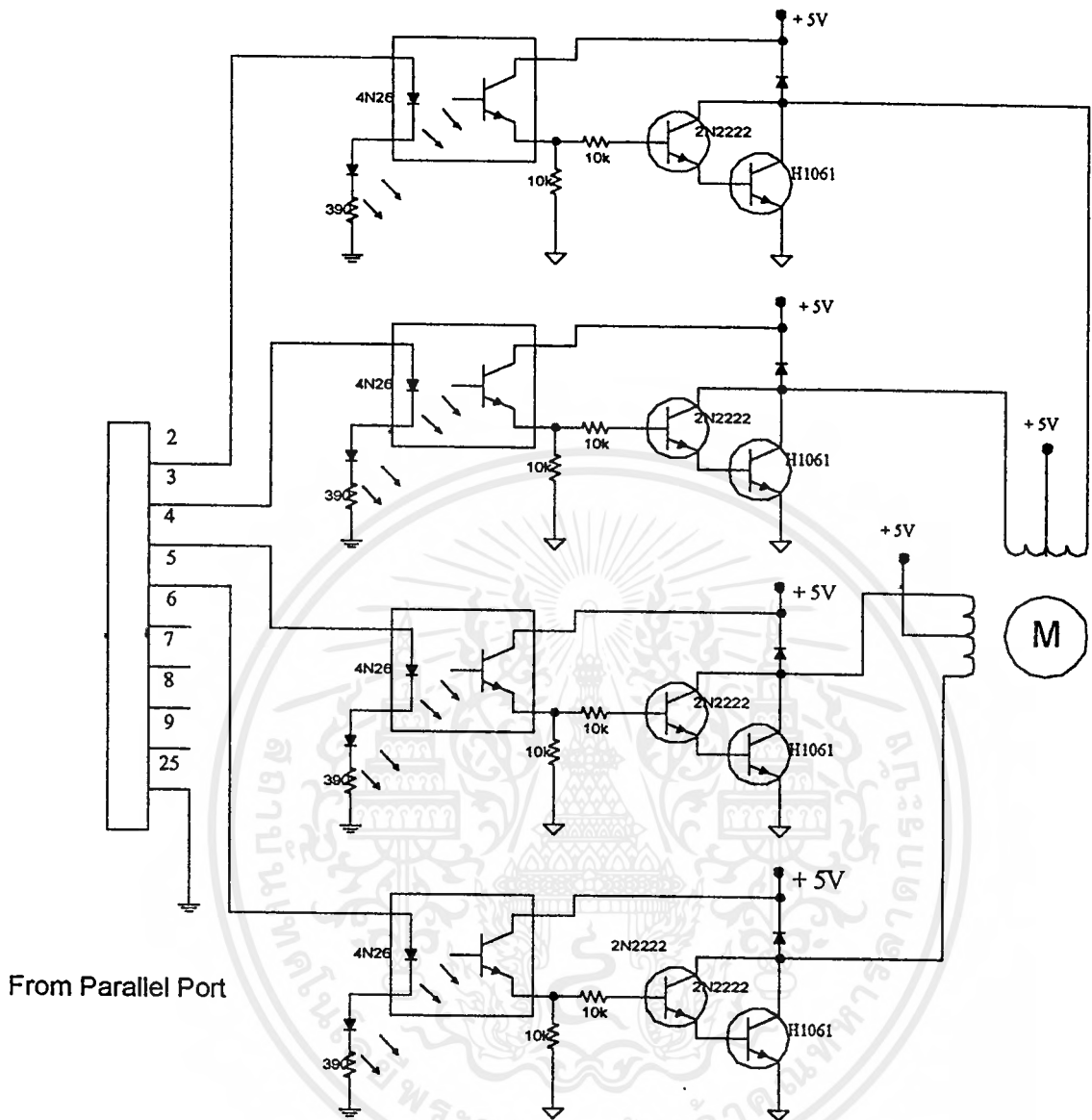


รูปที่ 6.2 แสดงบล็อกไดอะแกรมการทำงานของ Step

6.2.1 วงจร Drive Motor



รูปที่ 6.3 แสดงบล็อกไดอะแกรมของวงจร Drive Motor



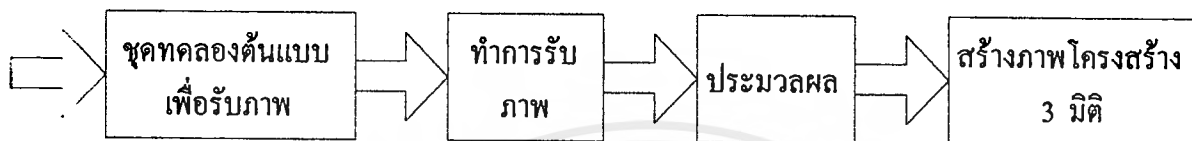
รูปที่ 6.4 แสดงวงจร Drive Motor

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 7

โปรแกรมการทำงาน

7.1 โครงสร้างการทำงานของกล้องถ่ายภาพ 3 มิติ



รูปที่ 7.1 แสดงการทำงานของกล้องถ่ายภาพ 3 มิติ

7.2 โครงสร้างของโปรแกรมการทำงาน

โปรแกรมการทำงานประกอบด้วย 3 ส่วนใหญ่ๆ ดังนี้

7.2.1 โปรแกรม Drive motor

- 1) รับข้อมูลที่ใช้ในการควบคุมวงจร Drive motor จากผู้ใช้
- 2) สร้างสัญญาณขับมอเตอร์จากข้อมูลที่ได้รับ
- 3) ส่งสัญญาณไปขับมอเตอร์และรอการถ่ายภาพ

7.2.2 โปรแกรมประมวลผล

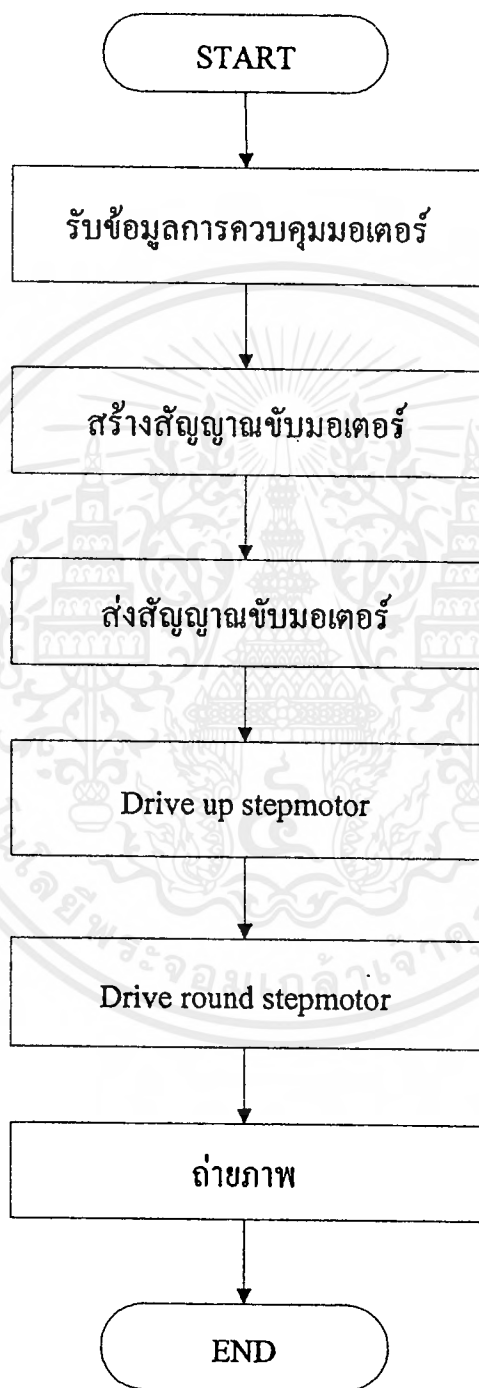
- 1) ข้อมูลจากภาพถ่ายจะถูกนำมาประมวลผล โดยมี 1 ข้อมูลต่อ 1 ภาพซึ่งค่าที่ได้เป็นระยะจากจุดที่สว่างที่สุด (จากแสงเลเซอร์มาตกบนวัตถุ) ถึงจุดศูนย์กลางของงานหมุน
- 2) นำข้อมูลที่ได้จากการประมวลผลเก็บไว้ในแฟ้มข้อมูลเพื่อนำไปสร้างภาพ 3 มิติ

7.2.3 โปรแกรมสร้างภาพ 3 มิติ

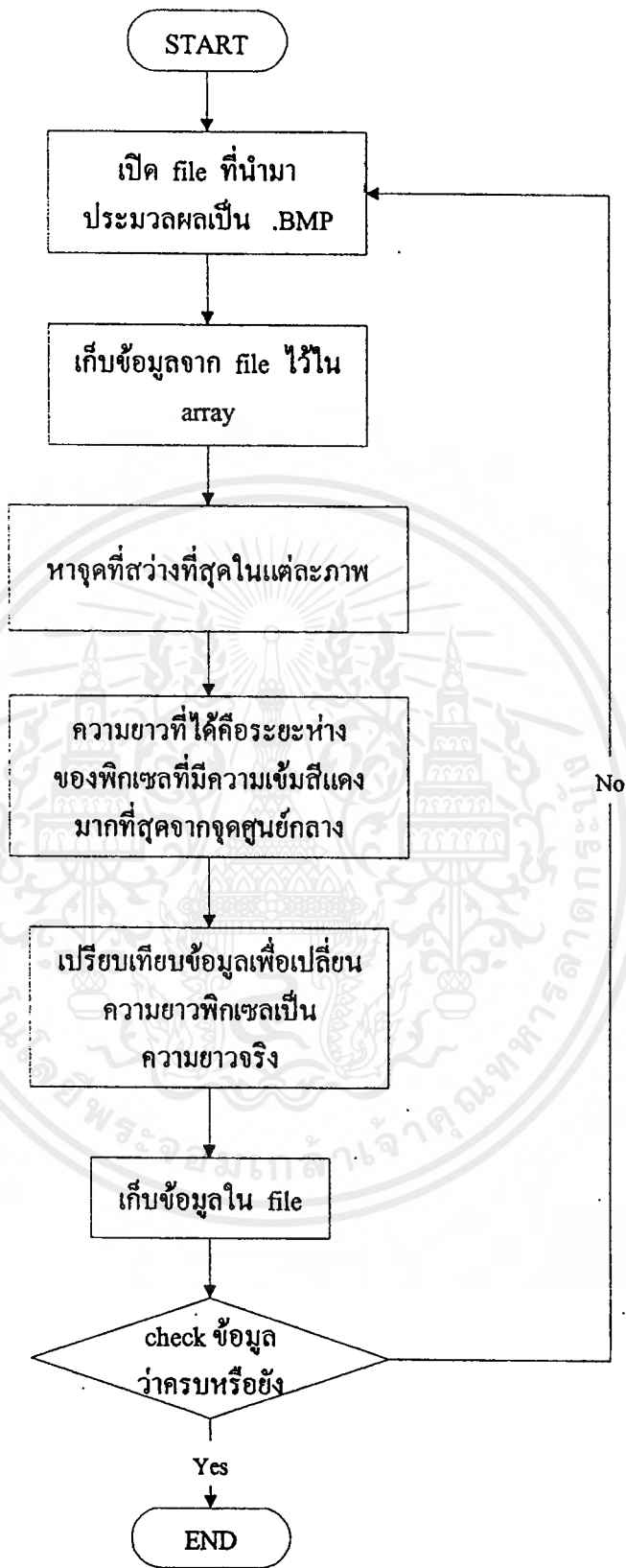
- 1) รับผลเข้ามาเป็น array ของ R และ Z (โดยที่ R เป็นระยะห่างของจุดเลเซอร์กับจุดศูนย์กลางของการหมุน และ Z เป็นความสูงของเลเซอร์เทียบกับงานหมุน)
- 2) นำระยะ R และ Z มาแปลงเป็นค่าในพิกัดคาร์ทีเซียน (x, y, z)
- 3) แปลงค่าจากพิกัดคาร์ทีเซียนให้เป็นพิกัดของรูป 3 มิติในระนาบ 2 มิติ
- 4) นำพิกัดที่คำนวณได้ไปพล็อตเป็นภาพ 3 มิติ
- 5) ถ้าต้องการทำการ Scaling, Rotation หรือ Translation ต้องทำการหาพิกัดคาร์ทีเซียนใหม่โดยใช้สมการการ Scaling, Rotation หรือ Translation ตามลำดับ แล้วนำพิกัดคาร์ที

เขียนที่ได้มาแปลงให้เป็นพิกัดของรูป 3 มิติในระนาบ 2 มิติ นำไปพล็อตได้เป็นภาพ 3 มิติ

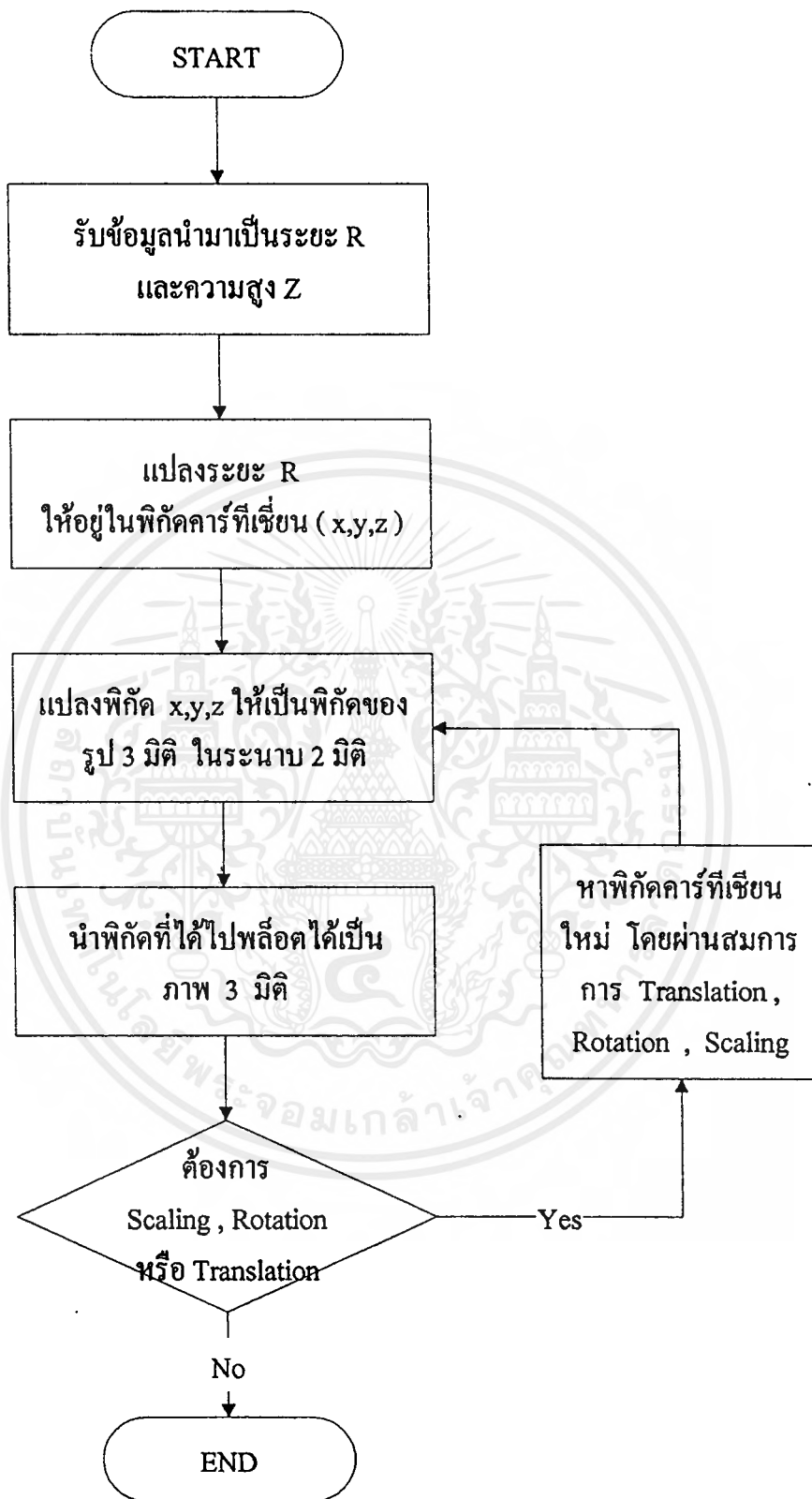
6) ถ้าไม่ต้องการทำอะไรต่อแล้ว ก็กดปุ่ม Exit เพื่อจบโปรแกรม



รูปที่ 7.2 แสดง Flow Chart ของโปรแกรม Drive motor



รูปที่ 7.3 แสดง Flow Chart ของ โปรแกรมประมวลผล



รูปที่ 7.4 แสดง Flow Chart ของ โปรแกรมสร้างภาพ 3 มิติ

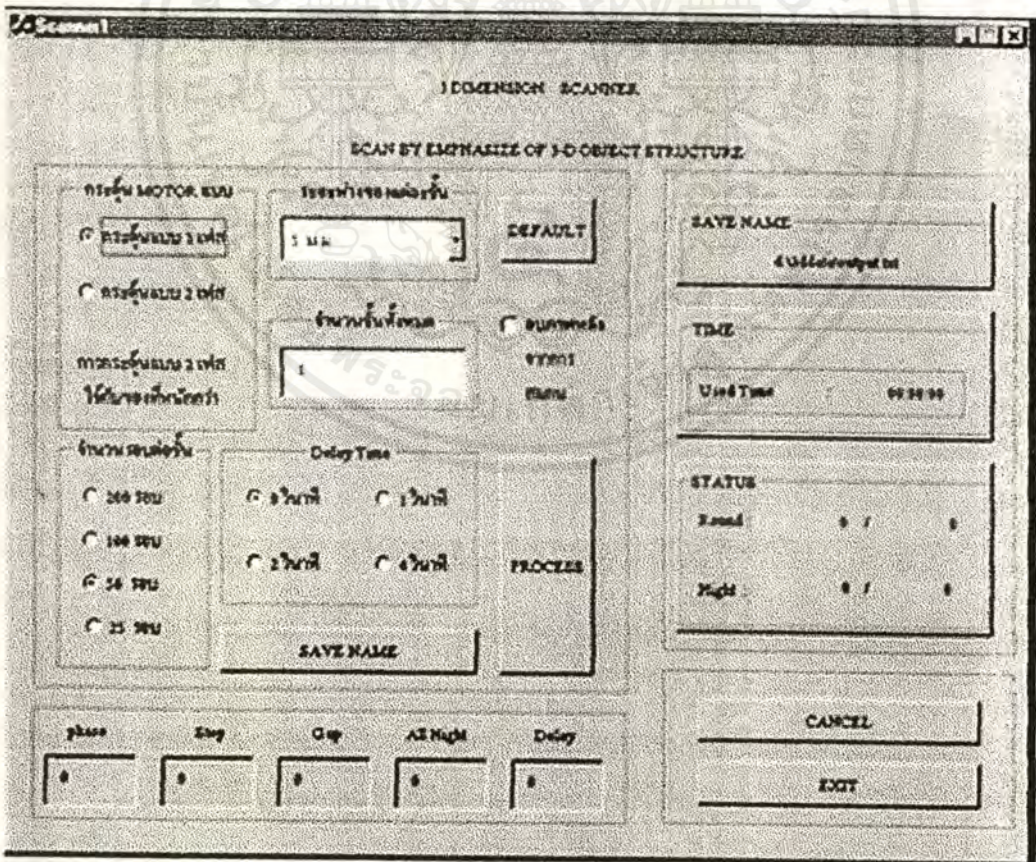
บทที่ 8

การทดลอง และผลการทดลอง

8.1 วิธีการทดลอง

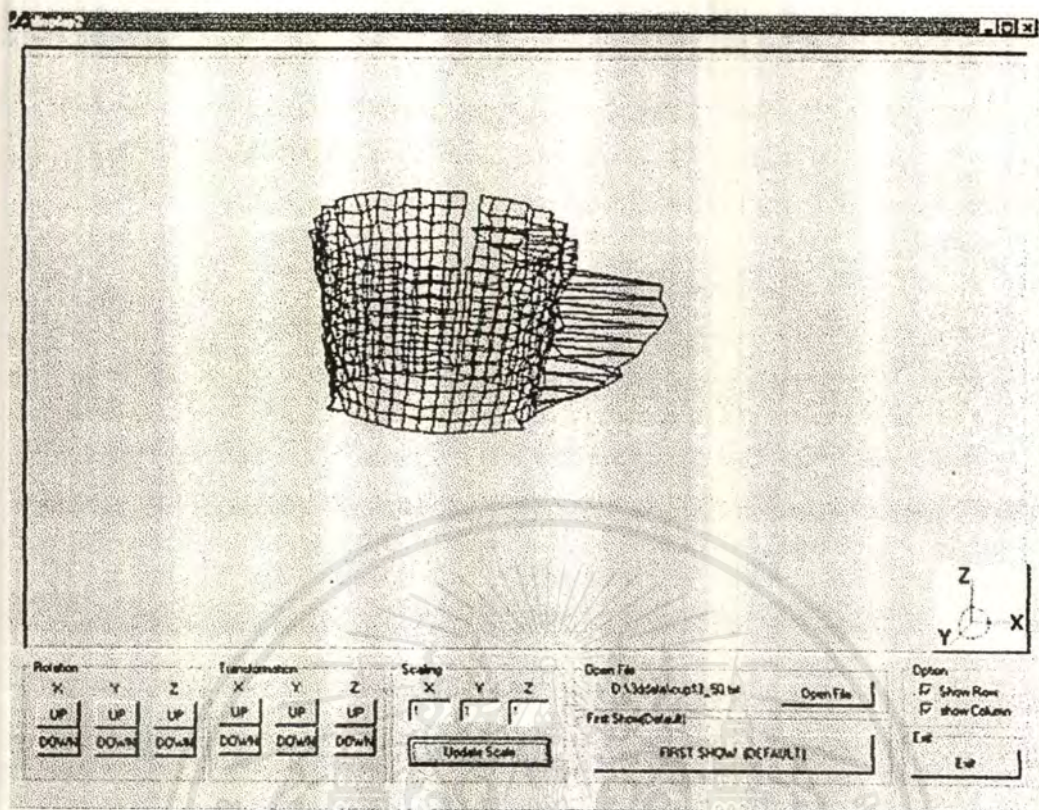
- 1) นำวัตถุที่จะนำมาสแกนวางบนจานหมุน โดยให้วัตถุที่บดบังกึ่งกลางของจานหมุน
- 2) ทำพื้นหลังมืด โดยการปิดฝาครอบเพื่อให้ง่ายในการประมวลผล
- 3) ทำการใส่ข้อมูลเพื่อการ Run Program
- 4) Run Program
- 5) ดูผลจากหน้าจอแสดงผล

8.2 ผลการทดลอง

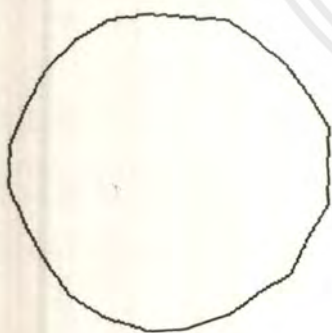


รูปที่ 8.1 แสดงหน้าจอรับข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 8.2 แสดงหน้าจอแสดงผล

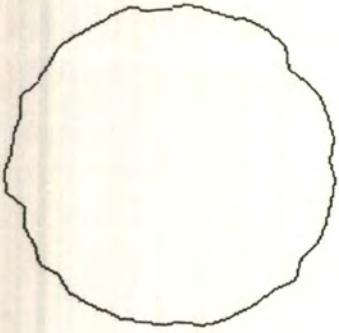


25 step/round Time 0.35 mins

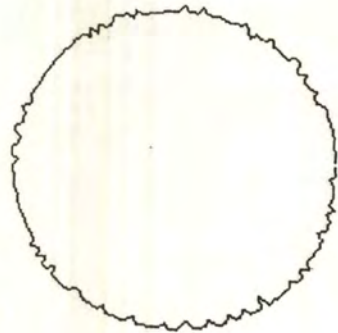


50 step/round Time 1.10 mins

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



100 step/round Time 2.15 mins



200 step/round Time 4.40 mins

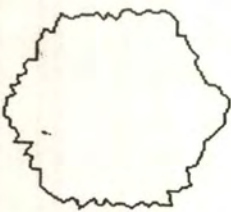
รูปที่ 8.3 แสดงภาพที่ได้จากการสแกนด้วยที่ 25,50,100,200 step/round



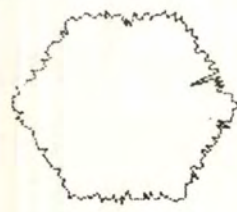
25 step/round Time 0.35 mins



50 step/round Time 1.10 mins



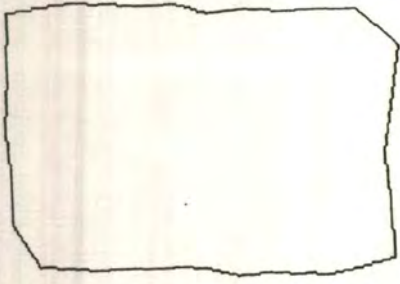
100 step/round Time 2.21 mins



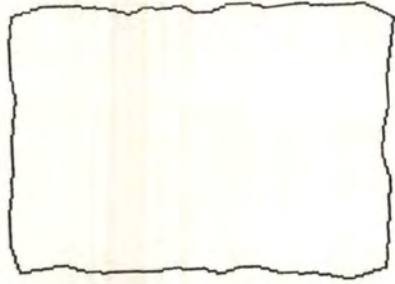
200 step/round Time 4.32 mins

รูปที่ 8.4 แสดงภาพที่ได้จากการสแกนปริซึมฐานหกเหลี่ยมที่ 25,50,100,200 step/round

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



25 step/round Time 0.35 mins



50 step/round Time 1.09 mins

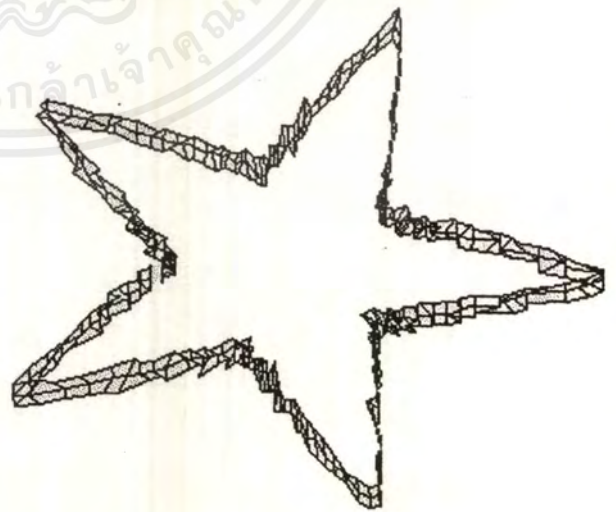


100 step/round Time 2.23 mins



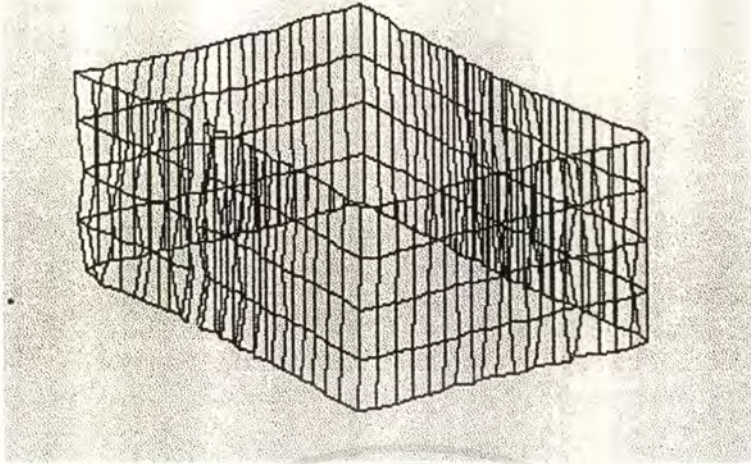
200 step/round Time 4.53 mins

รูปที่ 8.5 แสดงภาพที่ได้จากการสแกนปริซึมฐานสี่เหลี่ยมผืนผ้าที่ 25,50,100,200 step/round



รูปที่ 8.6 แสดงภาพที่ได้จากการสแกนปริซึมฐานรูปดาวที่ 200 step/round 1 และ 3 ชั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Time 12.10 mins

รูปที่ 8.7 แสดงภาพที่ได้จากการสแกนปริซึมฐานสี่เหลี่ยมผืนผ้า ที่ 100 step/round 5 ชั้น



(a)

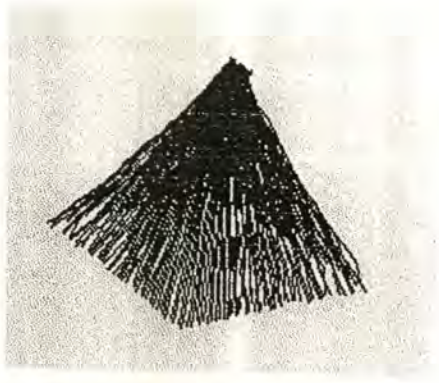
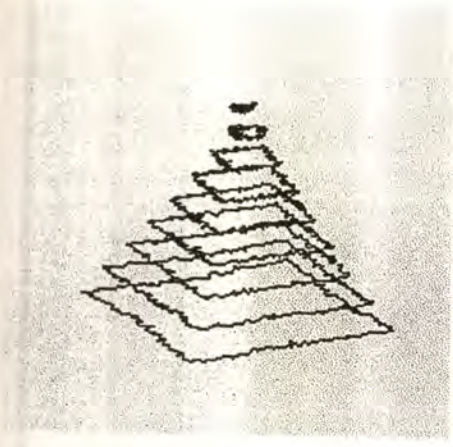
(b)

Time 46.24 mins

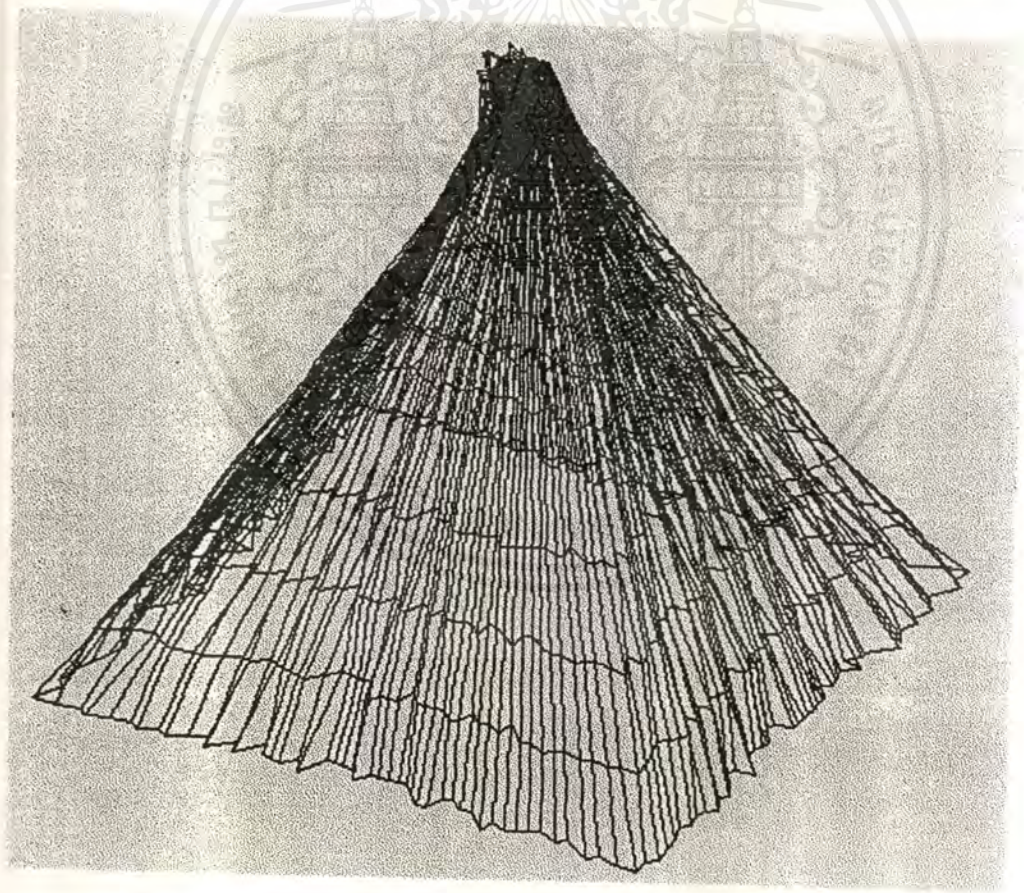
รูปที่ 8.8 (a) แสดงภาพที่ได้จากการสแกนปิระมิดฐานสี่เหลี่ยม ที่ 200 step/round 9 ชั้น

(b) แสดงภาพที่ได้จากการสแกนปิระมิดฐานสี่เหลี่ยมที่นำจุดผิดที่เกิดจากแสงไม่ตกกระทบผิววัตถุออก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

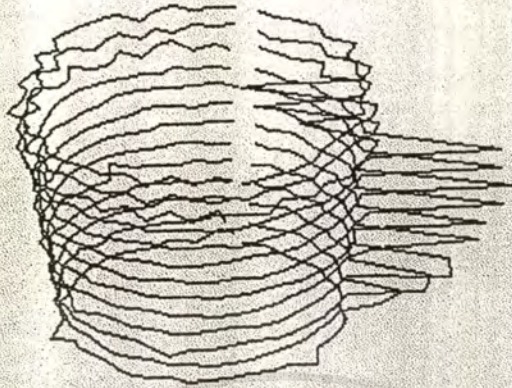


รูปที่ 8.9 แสดงภาพที่ได้จากการสแกนปริมาตรฐานสี่เหลี่ยมที่นำจุดผิดที่เกิดจากเมล็ดสีของกล้องออก



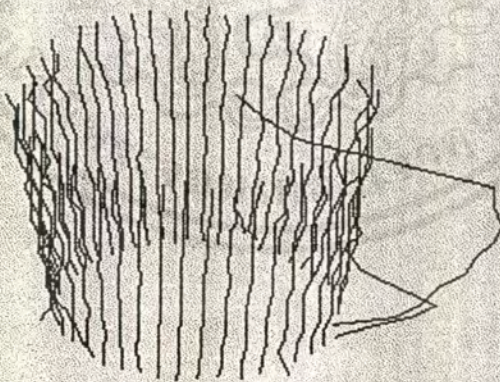
รูปที่ 8.10 แสดงภาพที่ได้จากการสแกนปริมาตรฐานสี่เหลี่ยมที่สมบูรณ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



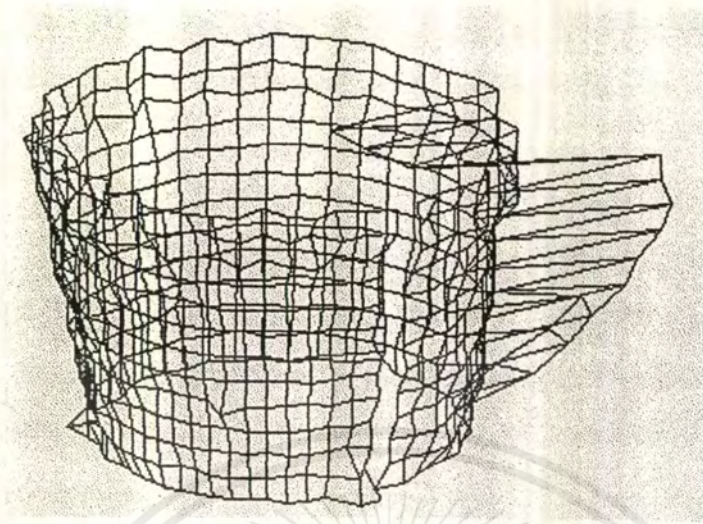
(a)

Time 10.36 mins



(b)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



(c)

รูปที่ 8.11 แสดงภาพที่ได้จากการสแกนแก้วกาแฟที่ 50 step/round 13 ชั้น

- (a) ภาศค้คขวง (แนวนอน)
- (b) ภาศค้คแนวตั้ง
- (c) ภาพโครงร่างตาข่าย

บทที่ 9

สรุปผลและวิจารณ์ผลการทดลอง

9.1 สรุปผลการทดลอง

ข้อมูลที่ได้เหมือนวัตถุจริงได้มากขึ้นมาก และการทำงานเร็วขึ้นเนื่องจากเป็นระบบอัตโนมัติทั้งหมด (กลุ่มเดียวสามารถทำงานได้แต่ต้นจนจบ) จากเทอมที่แล้วที่ต้องให้ผู้ใช้งานช่วยทำให้ใช้เวลาในการประมวลแต่ละภาพถึงเกือบ 4 ชั่วโมง แต่ในการสแกนของเทอมนี้ใช้เวลาเพียง ประมาณ 15 นาทีและยังให้จำนวนชั้นข้อมูลที่มากขึ้นด้วย ภาพที่ได้ยังสามารถเปลี่ยนรูปแบบของการแสดงผลได้ในหลายมุมมอง โดยผ่านกระบวนการ การเปลี่ยนจุดศูนย์กลางของภาพ , การหมุนภาพ , การขยายและย่อภาพ โดยทั้ง 3 กระบวนการนั้นสามารถทำได้ทุกแนวแกน

ตรงตามจุดประสงค์ที่ได้ตั้งไว้ที่จะเก็บข้อมูลเชิงโครงสร้างเพื่องานด้าน Image Processing ก็สามารถนำไปประยุกต์ทำงานได้จริง

9.2 วิจารณ์การทดลอง

1. ค่า STEP/ROUND ถ้าค่านี้มีค่ามาก ภาพที่ได้จะมีความถูกต้องมาก แต่เวลาที่ใช้ในการทดลองก็จะมากตามไปด้วย
2. จำนวนชั้นของการสแกน ถ้ามากภาพที่ได้จะมีความต่อเนื่องของด้านความสูงมากขึ้น แต่เวลาที่ใช้ในการทดลองก็จะมากตามไปด้วย
3. เวลาที่ใช้ในการทดลอง (การทำงาน บนเครื่อง PentiumII 400MHz Ram 128 Mb) โดยเฉลี่ย 0.35 นาที สำหรับ 25 STEP/ROUND , 1.10 นาที สำหรับ 50 STEP/ROUND , 2.21 นาที สำหรับ 100 STEP/ROUND และ 4.40 นาที สำหรับ 200 STEP/ROUND

รูปแบบ	25 STEP	50 STEP	100 STEP	200 STEP
รูป วงกลม/1 ชั้น	0.35	1.10	2.15	4.40
รูป หกเหลี่ยม/1ชั้น	0.35	1.10	2.21	4.32
รูป สี่เหลี่ยม/1ชั้น	0.35	1.09	2.23	4.53

4. รูปร่างของวัตถุที่สแกน รูปที่มีมุมแหลมๆ จะสแกนยากกว่า คือจะเกิดความเพี้ยนได้ง่ายกว่า เนื่องจากโอกาสที่แสงตกกระทบไม่พอดีกับมุมมีมาก จึงต้องใช้จำนวน step/round

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มากขึ้นเพื่อแก้ปัญหา นี้ เช่น ปริซึมฐานหกเหลี่ยมจะสแกนออกมาได้รูปเพี้ยนกว่าสามเหลี่ยมฐานสี่เหลี่ยมเพี้ยนกว่าปริซึมฐานหกเหลี่ยม และปริซึมฐานรูปดาวจะผิดเพี้ยนมากที่สุด

5. จากการทดลองรูปทรงปริระมิดมียอดแหลม ถ้าวางวัตถุในแนวที่ยอดแหลมไม่โดนแสงเลเซอร์จะทำให้ผลลัพธ์ที่ได้ผิดพลาด

จากสรุปผลการทดลองนี้เกิดข้อผิดพลาดบางประการทำให้ภาพที่ได้เกิดการผิดพลาด โดยจะแบ่งได้ดังนี้

9.2.1 วัตถุที่นำมาสแกน

1. สีของวัตถุ ถ้าวัตถุมีพื้นผิวสีดำ จะสามารถลดการสะท้อนของแสงเลเซอร์ที่ตกบนวัตถุ ทำให้รวมเป็นจุดได้ดี ส่วนสีขาวจะสะท้อนแสงทำให้แสงกระจายมาก
2. ความโค้งของวัตถุ ถ้าวัตถุมีความโค้งมากแล้วระยะห่างของแต่ละชั้นมีค่ามาก ข้อมูลที่ได้จะไม่มีคมชัดของด้านสูง ภาพที่ได้จะผิดเพี้ยน และการที่วัตถุมีความโค้งทำให้แสงตกกระทบไม่ตั้งฉากทำให้พื้นที่การตกกระทบเพิ่มขึ้น ความเข้มของเลเซอร์จะลดลง
3. วัตถุที่มีความใส เช่น แก้ว หรือวัตถุที่มีสีขาวมากจะเกิดการหักเหและสะท้อนแสงมาก จึงไม่เหมาะสำหรับการสแกน
4. ความสูงของวัตถุไม่เป็นจำนวนเท่าของระยะการหมุนขึ้น 1 สเตป เลเซอร์ที่สแกนในชั้นสุดท้ายจะเลยความสูงของวัตถุ ทำให้ข้อมูลผิดพลาด เครื่องสแกนเนอร์ในโรงงานจีนนี้ไม่สามารถ สแกนวัตถุที่เป็นวัตถุรูปเว้า(คอกขวด)เพราะจะต้องจะไม่สามารถจับตำแหน่งของเลเซอร์ได้เพราะถูกตัววัตถุเองบังไป และไม่สามารถจับข้อมูลที่เป็นวัตถุที่กลวงได้ เหตุผลหลักที่ไม่สามารถสแกนวัตถุพวกนี้ได้เพราะว่า ในโรงงานจีนนี้ใช้กล้องเพียงตัวเดียวเท่านั้น

9.2.2. HARDWARE

1. ความไม่แน่นอนของมอเตอร์ การทำงานของมอเตอร์ในการทำงานครั้งแรกบางทีมอเตอร์จะเกิดการกระตุก ทำให้มุมที่ได้ไม่เป็นไปตามมุมที่ตั้งไว้
2. เลเซอร์ ที่นำมาใช้ควรเป็นเลเซอร์ที่มีแสงเข้ม ให้จุดที่ได้เล็กและแสงไม่แตกกระจาย
2. กล้อง ต้องไม่มีจุดสีที่ค้างอยู่ในกล้อง กล้องที่ได้นำมาใช้นี้มีปัญหาอย่างมากกับจุดสี ถ้าความเข้มของแสงที่ตกลงบนวัตถุมีความเข้มน้อยกว่าความเข้มของเม็ดสี ทำให้

ประมวลผลผิดพลาด อีกปัญหาหนึ่งของกล้องคือ ถึงแม้ว่าจะถ่ายรูปเดียวกัน 2 ครั้งแต่ก็ไม่ได้หมายความว่า ภาพทั้งสองจะเหมือนกันทุกประการ ถึงแม้ว่าจะคงสภาพแสงและวัตถุเพียงใดก็ตาม

3. อุปกรณ์แมคคานิกส์เกิดเนื่องจากเกลียวหมุนไม่ตรง งานหมุนไม่ได้ขนานกับฐานจริง ทำให้วัตถุเอียง

9.2.3 จาก SOFTWARE

การหาแสงที่เข้มที่สุดที่ตกบนพื้นผิวของวัตถุก็ไม่แน่เสมอไปว่าจะได้ข้อมูลที่เป็นจุดศูนย์กลางของเลเซอร์ แต่จากการทดลองใช้วิธีหา ตำแหน่งของข้อมูลหลายๆแบบ ปรากฏว่าแบบที่ใช้อยู่ให้ความถูกต้องมากกว่าแบบอื่นๆ

9.3 แนวทางการพัฒนา

หาแนวคิดเสริมที่จะนำไปรวมเข้ากับแนวคิดหลักที่ผ่านมา ในการลดข้อจำกัดเรื่อง วัตถุที่นำมาสแกน รวมถึงเรื่อง สีของวัตถุ ความใส การสะท้อนแสง และความโค้งเว้า โดยอาจจะเพิ่มกล้องเข้าไปอีกหนึ่งตัว หรือหาอัลกอริทึมทางซอฟต์แวร์ในการแก้ไขปัญหา

แนวทางการแก้เรื่องความสูงของวัตถุให้ใช้วิธี แสแกนจากบนลงล่างแล้วเมื่อถึงขั้นสุดท้าย (ชั้นล่างสุด) ก็จะมีตัวเซ็นเซอร์ที่ไว้ตรวจจับว่าความสูงนี้เป็นความสูงสุดท้ายที่จะเป็นไปได้ และส่งข้อมูลกลับไปให้ คอมพิวเตอร์ คอมพิวเตอร์จะทำการหยุดส่งมอเตอร์ แล้วสแกนชั้นนั้นเป็นขั้นสุดท้ายแทน คอมพิวเตอร์จะประมวลผลว่าชั้นที่สแกนนั้นห่างจากชั้นก่อนหน้าเป็นระยะทางเท่าไรโดยคำนวณจากจำนวนสเตปของมอเตอร์ที่หมุนไปแล้ว แต่ในโครงการนี้ เราได้ใช้ PORT สำหรับส่งข้อมูลทั้งหมด (8 สัญญาณ) ดังนั้นถ้าจะต้องการส่ง Feedback กลับไปให้คอมพิวเตอร์จึงต้องการสายสัญญาณข้อมูลอีก 1 สาย ดังนั้นจึงต้องสร้างPORT ข้อมูลขึ้นมาอีก 1 PORT เพื่อใช้ในการส่งข้อมูลอีก 1 สายสัญญาณ

โปรแกรมสำหรับสร้างความต่อเนื่องให้ข้อมูล จากปัญหาที่พบเรื่องข้อมูลไม่ได้เป็นเส้นตรงที่ตรงจริงๆ หรือได้วงกลมที่เส้นรอบวงไม่เรียบ ดังนั้นจึงควรสร้าง โปรแกรมที่จะช่วยในการสร้างความต่อเนื่องของข้อมูลอาจจะเป็นทางสถิติ หรือ อาจเป็นทาง Image processing

เอกสารอ้างอิง

1. ชันวา ศรีประมง, “ การเขียนโปรแกรมภาษาซีสำหรับวิศวกรรม ”, โครงการตำราวิชาการ มหาวิทยาลัยเทคโนโลยีมหานคร , 2521.
2. ประภาส บวรปัญญา, “ การโปรแกรมแบบเน้นวัตถุ C++ ”, หจก. สำนักพิมพ์ฟิสิกส์เซ็นเตอร์.
3. มัณฑนา ปราการสมุทร, “ การเขียนชุดคำสั่งภาษา C ”, คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย , 2534 .
4. สานนท์ เคลือบกำเหนิด , “ คู่มือการใช้ Microsoft Visual C++ เวอร์ชัน 2.0 ” , บริษัท ซีเอ็ดยูเคชั่น จำกัด (มหาชน) , 2539.
5. อนิรุต ถิวหาทอง , “ การเขียนโปรแกรมบนวินโดวส์ด้วย Microsoft C++ ” , บริษัท ซีเอ็ดยูเคชั่น จำกัด (มหาชน) , 2539.
6. Davis Chapman , “ Sams Teach Yourself Visual C++ 6 in 21 Days ” , Sams Publishing , 1998.
7. Ori Gurewich and Nathan Gurewich , “ Teach Yourself Visual C++ 4 in 21 Days ” , Sams Publishing , 1996.
8. , บริษัท ซีเอ็ดยูเคชั่น จำกัด (มหาชน) , รวมโครงการ อิเล็กทรอนิกส์เครื่องควบคุมมอเตอร์, บริษัท ซีเอ็ดยูเคชั่น จำกัด (มหาชน) , 2539.
9. <http://www.quickcom.com>
10. <http://kolban.com/video/>

ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SCANNER

จากfile Scanner1Dlg.cpp

```
// Scanner1Dlg.cpp : implementation file
```

```
//
```

```
#include "stdafx.h"
```

```
//ส่วนเริ่ม include header
```

```
#include "Scanner1.h"
```

```
#include "Scanner1Dlg.h"
```

```
#include <conio.h>
```

```
#include <vidcap.h>
```

```
#include <stdio.h>
```

```
#include <iostream.h>
```

```
#include <conio.h>
```

```
//ส่วนจบ include header
```

```
#ifdef _DEBUG
```

```
#define new DEBUG_NEW
```

```
#undef THIS_FILE
```

```
static char THIS_FILE[] = __FILE__;
```

```
#endif
```

```
//ส่วนเริ่มที่ Visual C generate ขึ้นมา
```

```
//ส่วนจบที่ Visual C generate ขึ้นมา
```

```
char PicName[]="D:\\photo\\h00f000.bmp";
```

```
//ส่วนเริ่มประกาศตัวแปล
```

```
int OldNext=0;
```

```
int CountRound=0;
```

```
int next;
```

```
char OutRed100,OutRed10,OutRed1;
```

```
int meter,Hundred;
```

```
int hour,minute,second;
```

```
int oldsec;
```

```
int intersec;
```

```
int firsttime;
```

```
int usetime;
```

```
int PunchH;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

int pr1,pr2,pr3,pr4;
int ph1,ph2,ph3,ph4;
int delFactor; //ส่วนจบประกาศตัวแปล

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
//ส่วนเริ่ม aboutbox
// CAboutDlg dialog used for App About

class CAboutDlg : public CDialog
{
public:
    CAboutDlg();

// Dialog Data
//{{AFX_DATA(CAboutDlg)
enum { IDD = IDD_ABOUTBOX };
//}}AFX_DATA

// ClassWizard generated virtual function overrides
//{{AFX_VIRTUAL(CAboutDlg)
protected:
virtual void DoDataExchange(CDataExchange* pDX); // DDX/DDV support
//}}AFX_VIRTUAL

// Implementation
protected:

//{{AFX_MSG(CAboutDlg)
//}}AFX_MSG
DECLARE_MESSAGE_MAP()
};

CAboutDlg::CAboutDlg() : CDialog(CAboutDlg::IDD)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
//{{AFX_DATA_INIT(CAboutDlg)
```

```
//}}AFX_DATA_INIT
```

```
void CAboutDlg::DoDataExchange(CDataExchange* pDX)
```

```
{
    CDialog::DoDataExchange(pDX);
```

```
//{{AFX_DATA_MAP(CAboutDlg)
```

```
//}}AFX_DATA_MAP
```

```
BEGIN_MESSAGE_MAP(CAboutDlg, CDialog)
```

```
//{{AFX_MSG_MAP(CAboutDlg)
```

```
    // No message handlers
```

```
//}}AFX_MSG_MAP
```

```
END_MESSAGE_MAP()
```

```
//ส่วนจบaboutbox
```

```
////////////////////////////////////
```

```
// CScanner1Dlg dialog
```

```
//Visual C generate
```

```
CScanner1Dlg::CScanner1Dlg(CWnd* pParent /*=NULL*/)
    : CDialog(CScanner1Dlg::IDD, pParent)
```

```
{
    {{{AFX_DATA_INIT(CScanner1Dlg)
```

```
m_iPhase = -1;
```

```
m_iDelay = -1;
```

```
m_iStep = -1;
```

```
m_strGap = _T("");
```

```
m_sName = _T("");
```

```
m_bDel = FALSE;
```

```
m_iDelayR = 0;
```

```
m_iGapR = 0;
```

```
m_iPhaseR = 0;
```

```
m_iStepR = 0;
```

```
m_iAllhightR = 0;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

m_bDelR = FALSE;
Ok = 0;
m_iTotalRound = 0;
m_iAllhight = 0;
m_iNewRound = 0;
m_sTime = _T("");
m_iNewHight = 0;
m_iTotalHight = 0;
//}}AFX_DATA_INIT
// Note that LoadIcon does not require a subsequent DestroyIcon in Win32
m_hIcon = AfxGetApp()->LoadIcon(IDR_MAINFRAME);
}

```

```

void CScanner1Dlg::DoDataExchange(CDataExchange* pDX)
{

```

```

    CDialog::DoDataExchange(pDX);
    //{{AFX_DATA_MAP(CScanner1Dlg)
    DDX_Radio(pDX, IDC_PHASE1, m_iPhase);
    DDX_Radio(pDX, IDC_DELAY1, m_iDelay);
    DDX_Radio(pDX, IDC_STEP1, m_iStep);
    DDX_CBString(pDX, IDC_GAP, m_strGap);
    DDX_Text(pDX, IDC_FILENAME, m_sName);
    DDX_Check(pDX, IDC_DEL, m_bDel);
    DDX_Text(pDX, IDC_SHOWDELAY, m_iDelayR);
    DDX_Text(pDX, IDC_SHOWGAP, m_iGapR);
    DDX_Text(pDX, IDC_SHOWPHASE, m_iPhaseR);
    DDX_Text(pDX, IDC_SHOWSTEP, m_iStepR);
    DDX_Text(pDX, IDC_SHOWALLHIGHT, m_iAllhightR);
    DDX_Text(pDX, IDC_TOTALROUND, m_iTotalRound);
    DDX_Text(pDX, IDC_ALLHIGHT, m_iAllhight);
    DDX_Text(pDX, IDC_NEWROUND, m_iNewRound);
    DDX_Text(pDX, IDC_USETIME, m_sTime);
    DDX_Text(pDX, IDC_NEWHIGHT, m_iNewHight);
    DDX_Text(pDX, IDC_TOTALHIGHT, m_iTotalHight);
    //}}AFX_DATA_MAP

```

```

}
BEGIN_MESSAGE_MAP(CScanner1Dlg, CDialog)
//{{AFX_MSG_MAP(CScanner1Dlg)
ON_WM_SYSCOMMAND()
ON_WM_PAINT()
ON_WM_QUERYDRAGICON()
ON_BN_CLICKED(IDC_EXIT, OnExit)
ON_BN_CLICKED(IDC_PROCESS, OnProcess)
ON_BN_CLICKED(IDC_CAN, OnCan)
ON_BN_CLICKED(IDC_DEFAULT, OnDefault)
ON_BN_CLICKED(IDC_SAVENAME, OnSavename)
ON_WM_TIMER()
ON_BN_CLICKED(IDC_DEL, OnDel)
//}}AFX_MSG_MAP
END_MESSAGE_MAP()

////////////////////////////////////
// CScanner1Dlg message handlers

BOOL CScanner1Dlg::OnInitDialog()
{
    CDialog::OnInitDialog();

    // Add "About..." menu item to system menu.

    // IDM_ABOUTBOX must be in the system command range.
    ASSERT((IDM_ABOUTBOX & 0xFFFF) == IDM_ABOUTBOX);
    ASSERT(IDM_ABOUTBOX < 0xF000);

    CMenu* pSysMenu = GetSystemMenu(FALSE);
    if (pSysMenu != NULL)
    {
        CString strAboutMenu;
        strAboutMenu.LoadString(IDS_ABOUTBOX);
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if (!strAboutMenu.IsEmpty())
{
    pSysMenu->AppendMenu(MF_SEPARATOR);
    pSysMenu->AppendMenu(MF_STRING, IDM_ABOUTBOX, strAboutMenu);
}
}

```

// Set the icon for this dialog. The framework does this automatically

// when the application's main window is not a dialog

SetIcon(m_hIcon, TRUE); // Set big icon

SetIcon(m_hIcon, FALSE); // Set small icon

// TODO: Add extra initialization here //ตั้งค่าตั้งต้น สำหรับ ช่องกรอกต่างๆ

//Set START PARAMETER

m_bDel=FALSE;

m_iDelay=0;

m_iStep=2;

m_iPhase=0;

m_iAllhigh=1;

m_strGap=" 5 □□.:";

m_sTime="00:00:00";

m_sName="d:\\3ddata\\output.txt";

_outpw(0x378,0);

UpdateData(FALSE);

//จบตั้งค่าตั้งต้น สำหรับ ช่องกรอกต่างๆ

return TRUE; // return TRUE unless you set the focus to a control

void CScanner1Dlg::OnSysCommand(UINT nID, LPARAM lParam)

{

if ((nID & 0xFFF0) == IDM_ABOUTBOX)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{
    CAboutDlg dlgAbout;
    dlgAbout.DoModal();
}
else
{
    CDialog::OnSysCommand(nID, lParam);
}
}

```

// If you add a minimize button to your dialog, you will need the code below
// to draw the icon. For MFC applications using the document/view model,
// this is automatically done for you by the framework.

```

void CScanner1Dlg::OnPaint()
{
    if (IsIconic())
    {
        CPaintDC dc(this); // device context for painting

        SendMessage(WM_ICONERASEBKGND, (LPARAM) dc.GetSafeHdc(), 0);

        // Center icon in client rectangle
        int cxIcon = GetSystemMetrics(SM_CXICON);
        int cyIcon = GetSystemMetrics(SM_CYICON);
        CRect rect;
        GetClientRect(&rect);
        int x = (rect.Width() - cxIcon + 1) / 2;
        int y = (rect.Height() - cyIcon + 1) / 2;

        // Draw the icon
        dc.DrawIcon(x, y, m_hIcon);
    }
    else
    {

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
CDialog::OnPaint();
```

```
}
```

```
// The system calls this to obtain the cursor to display while the user drags
```

```
// the minimized window.
```

```
HCURSOR CScanner1Dlg::OnQueryDragIcon()
```

```
{
    return (HCURSOR) m_hIcon;
}
```

```
void CScanner1Dlg::OnExit()
```

```
{
    // TODO: Add your control notification handler code here
```

```
_outpw(0x378,0);
```

```
OnOK();
}
```

```
void CScanner1Dlg::OnCan()
```

```
{
    // TODO: Add your control notification handler code here
```

```
_outpw(0x378,0);
```

```
OnCancel();
}
```

```
void CScanner1Dlg::OnDefault()
```

```
{
    // TODO: Add your control notification handler code here
```

```
m_bDel=FALSE;
```

```
m_iDelay=0;
```

```
m_iStep=2;
```

```
m_iPhase=0;
```

```
m_iAllhight=1;
}
```



```

m_strGap=" 5  ";
m_sTime="00:00:00";
m_sName="d:\\3ddata\\output.txt";
UpdateData(FALSE);
}

```

```
void CScanner1Dlg::OnSavename()
```

```

{
    // TODO: Add your control notification handler code here
    //Choose file name to save form source code page91 listing 5.2
    //form VISUAL C++ 21 Days Version6
    //CString m_sName;
    UpdateData(TRUE);
    CFileDialog m_IdFile(FALSE,"111","d:\\3ddata\\output.txt",OFN_HIDEREADONLY |
    OFN_OVERWRITEPROMPT,
        "TXT File (*.txt)|*.txt|3Ddata File (*.3dd)|*.3dd|All Files (*.*)|*.*|");
    if(m_IdFile.DoModal()==IDOK)
    {
        m_sName=m_IdFile.GetPathName();
        m_IdFile.GetStartPosition();
        UpdateData(FALSE);
    }
}

```

```
void CScanner1Dlg::OnProcess()
```

```

{
    // TODO: Add your control notification handler code here
    remove("d:\\photo\\output.txt");
    usetime=0;
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
firsttime=0;
```

```
//DISABLE INPUT DIALOG
```

```
GetDlgItem(IDC_PHASE1) ->EnableWindow(FALSE);
```

```
GetDlgItem(IDC_PHASE2) ->EnableWindow(FALSE);
```

```
GetDlgItem(IDC_STEP1) ->EnableWindow(FALSE);
```

```
GetDlgItem(IDC_STEP2) ->EnableWindow(FALSE);
```

```
GetDlgItem(IDC_STEP3) ->EnableWindow(FALSE);
```

```
GetDlgItem(IDC_STEP4) ->EnableWindow(FALSE);
```

```
GetDlgItem(IDC_DELAY1) ->EnableWindow(FALSE);
```

```
GetDlgItem(IDC_DELAY2) ->EnableWindow(FALSE);
```

```
GetDlgItem(IDC_DELAY3) ->EnableWindow(FALSE);
```

```
GetDlgItem(IDC_DELAY4) ->EnableWindow(FALSE);
```

```
GetDlgItem(IDC_GAP) ->EnableWindow(FALSE);
```

```
GetDlgItem(IDC_ALLHIGHT)->EnableWindow(FALSE);
```

```
//GetDlgItem(IDC_SETUP) ->EnableWindow(FALSE);
```

```
GetDlgItem(IDC_SAVENAME)->EnableWindow(FALSE);
```

```
//GetDlgItem(IDC_TOPVIEW) ->EnableWindow(FALSE);
```

```
GetDlgItem(IDC_DEFAULT) ->EnableWindow(FALSE);
```

```
GetDlgItem(IDC_DEL) ->EnableWindow(FALSE);
```

```
GetDlgItem(IDC_PROCESS) ->EnableWindow(FALSE);
```

```
//SET VARIABLE TO USE
```

```
UpdateData(TRUE);
```

```
//int m_iPhaseR;
```

```
if remove EDIT BOX
```

```
(IDC_SHOWPHASE)that have
```

```
//int m_iStepR;
```

variable name of m_iPhaseR

you will VAR this

```
//int m_iGapR;
```

variable here

```
//int m_iAllhightR;
```

```
//int m_iDelayR;
```

```
//int m_bDeIR;
```

```
//SET VARIABLE FOR m_iPhase
```

```
switch(m_iPhase)
```

```
{
```

```
case 0:
```

```
    m_iPhaseR=1;
```

```
    break;
```

```
case 1:
```

```
    m_iPhaseR=2;
```

```
    break;
```

```
}
```

```
//SET VARIABLE FOR m_iStep
```

```
switch(m_iStep)
```

```
{
```

```
case 0:
```

```
    m_iStepR=200;
```

```
    break;
```

```
case 1:
```

```
    m_iStepR=100;
```

```
    break;
```

```
case 2:
```

```
    m_iStepR=50;
```

```
    break;
```

```
case 3:
```

```
    m_iStepR=25;
```

```
    break;
```

```
}
```

```

//SET VARIABLE FOR m_iDelay
switch(m_iDelay)
{
case 0:
    m_iDelayR=0;
    break;

case 1:
    m_iDelayR=1;
    break;

case 2:
    m_iDelayR=2;
    break;

case 3:
    m_iDelayR=4;
    break;
}

//SET VARIABLE FOR m_strGap(m_strGap is CSTRING cannot use case method)
if(m_strGap==" 1 □□.")
{
    m_iGapR=1;
    PunchH=9;
}
if(m_strGap==" 2 □□.")
{
    m_iGapR=2;
    PunchH=17;
}
if(m_strGap==" 3 □□.")
{
    m_iGapR=3;
    PunchH=25;
}
if(m_strGap==" 4 □□.")
{

```

```

        m_iGapR=4;
        PunchH=34;
    }
    if(m_strGap==" 5 □□.")
    {
        m_iGapR=5;
        PunchH=42;
    }
    if(m_strGap==" 6 □□.")
    {
        m_iGapR=6;
        PunchH=50;
    }
    if(m_strGap==" 7 □□.")
    {
        m_iGapR=7;
        PunchH=59;
    }
    if(m_strGap==" 8 □□.")
    {
        m_iGapR=8;
        PunchH=67;
    }
    if(m_strGap==" 9 □□.")
    {
        m_iGapR=9;
        PunchH=75;
    }
    if(m_strGap=="10 □□.")
    {
        m_iGapR=10;
        PunchH=84;
    }
    if(m_strGap=="15 □□.")
    {

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        m_iGapR=15;
        PunchH=125;
    }
    if(m_strGap=="20 □□.")
    {
        m_iGapR=20;
        PunchH=167;
    }
    if(m_strGap=="50 □□.")
    {
        m_iGapR=50;
        PunchH=417;
    }

```

```
//SET VARIABLE FOR m_Allhight
```

```
m_iAllhightR=m_iAllhight;
```

```
//SetVARIABLE FOR m_bDel(m_bDel=Bool but if u VAR m_bDelR as bool
```

```
//it have a warning it must set to int
```

```
m_bDelR=m_bDel;
```

```
//if TRUE the DATA IN IDC_SHOW..... WILL BE TAKE TO m_i.....R
```

```
UpdateData(FALSE);
```

```
//Write header to output.txt
```

```
FILE *OutputWrite;
```

```
OutputWrite=fopen("D:\\photo\\output.txt","a");
```

```
switch(m_iStepR)
```

```
{
```

```
case 25:          fprintf(OutputWrite,"025");break;
```

```
case 50:          fprintf(OutputWrite,"050");break;
```

```
case 100:         fprintf(OutputWrite,"100");break;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

case 200:          fprintf(OutputWrite,"200");break;
default:          MessageBox("ERROR WRITE m_iStepR to Output.txt");
}

fprintf(OutputWrite," ");

switch(m_iAllhightR)
{
case 1:           fprintf(OutputWrite,"01");break;
case 2:           fprintf(OutputWrite,"02");break;
case 3:           fprintf(OutputWrite,"03");break;
case 4:           fprintf(OutputWrite,"04");break;
case 5:           fprintf(OutputWrite,"05");break;
case 6:           fprintf(OutputWrite,"06");break;
case 7:           fprintf(OutputWrite,"07");break;
case 8:           fprintf(OutputWrite,"08");break;
case 9:           fprintf(OutputWrite,"09");break;
case 10:          fprintf(OutputWrite,"10");break;
case 11:          fprintf(OutputWrite,"11");break;
case 12:          fprintf(OutputWrite,"12");break;
case 13:          fprintf(OutputWrite,"13");break;
case 14:          fprintf(OutputWrite,"14");break;
case 15:          fprintf(OutputWrite,"15");break;
case 16:          fprintf(OutputWrite,"16");break;
case 17:          fprintf(OutputWrite,"17");break;
case 18:          fprintf(OutputWrite,"18");break;
case 19:          fprintf(OutputWrite,"19");break;
case 20:          fprintf(OutputWrite,"20");break;
case 21:          fprintf(OutputWrite,"21");break;
case 22:          fprintf(OutputWrite,"22");break;
case 23:          fprintf(OutputWrite,"23");break;
case 24:          fprintf(OutputWrite,"24");break;
case 25:          fprintf(OutputWrite,"25");break;
case 26:          fprintf(OutputWrite,"26");break;
case 27:          fprintf(OutputWrite,"27");break;

```

```

case 28:          fprintf(OutputWrite,"28");break;
case 29:          fprintf(OutputWrite,"29");break;
case 30:          fprintf(OutputWrite,"30");break;
default :        MessageBox("ERROR WRITE m_iAllhighR to Output.txt");
}

```

```
fprintf(OutputWrite," ");
```

```
switch(m_iGapR)
```

```

{
case 1:          fprintf(OutputWrite,"01");break;
case 2:          fprintf(OutputWrite,"02");break;
case 3:          fprintf(OutputWrite,"03");break;
case 4:          fprintf(OutputWrite,"04");break;
case 5:          fprintf(OutputWrite,"05");break;
case 6:          fprintf(OutputWrite,"06");break;
case 7:          fprintf(OutputWrite,"07");break;
case 8:          fprintf(OutputWrite,"08");break;
case 9:          fprintf(OutputWrite,"09");break;
case 10:         fprintf(OutputWrite,"10");break;
case 15:         fprintf(OutputWrite,"15");break;
case 20:         fprintf(OutputWrite,"20");break;
case 50:         fprintf(OutputWrite,"50");break;
default :        MessageBox("ERROR WRITE m_iGapR to Output.txt");
}

```

```
fprintf(OutputWrite," ");
```

```
fclose(OutputWrite);
```

```
//start Drive step
```

```
//choose 1 Phase,2 Phase
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

switch(m_iPhaseR)
{
case 1:
    pr1= 16;//0001 0000
    pr2= 32;//0010 0000
    pr3= 64;//0100 0000
    pr4= 128;//1000 0000
    ph1= 136;//1000 1000
    ph2= 132;//1000 0100
    ph3= 130;//1000 0010
    ph4= 129;//1000 0001
    break;

case 2:
    pr1= 144;//1001 0000
    pr2= 48;//0011 0000
    pr3= 96;//0110 0000
    pr4= 192;//1100 0000
    ph1= 200;//1100 1000
    ph2= 196;//1100 0100
    ph3= 194;//1100 0010
    ph4= 193;//1100 0001
    break;

default :
    MessageBox("ERROR with punch(not 1,2 Phase)");
}

for(int h=1;h<=m_iAllheightR;h++) //ส่วนตั้งงานหมุน motor
{

    if(h!=1)
    {
        for(int y=1;y<=PunchH;y++)
        {
            for(int y1=1;y1<=10000;y1++)
                _outpw(0x378,ph1);
            for(int y2=1;y2<=10000;y2++)
                _outpw(0x378,ph2);
        }
    }
}

```

```

for(int y3=1;y3<=10000;y3++)
    _outpw(0x378,ph3);
for(int y4=1;y4<=10000;y4++)
    _outpw(0x378,ph4);
}
}

```

```

for(int r=1;r<=25;r++) //ส่วนการหยุดรับข้อมูล
{

```

```

//1
if(m_iStepR==200)
{

```

```

    CountTime();cap(h);
}

```

```

if(m_iStepR==100)
{

```

```

    CountTime();cap(h);
}

```

```

if(m_iStepR==50)
{

```

```

    CountTime();cap(h);
}

```

```

if(m_iStepR==25)
{

```

```

    CountTime();cap(h);
}

```

```

for(int x1=1;x1<=6000;x1++)

```

```

    _outpw(0x378,pr1);

```

```

//2

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if(m_iStepR==200)
{
    CountTime();cap(h);
}

for(int x2=1;x2<=6000;x2++)
    _outpw(0x378,pr2);

//3
if(m_iStepR==200)
{
    CountTime();cap(h);
}
if(m_iStepR==100)
{
    CountTime();cap(h);
}
for(int x3=1;x3<=6000;x3++)
    _outpw(0x378,pr3);

//4
if(m_iStepR==200)
{
    CountTime();cap(h);
}
for(int x4=1;x4<=6000;x4++)
    _outpw(0x378,pr4);

//5
if(m_iStepR==200)
{
    CountTime();cap(h);
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}
if(m_iStepR==100)
{
    CountTime();cap(h);
}
if(m_iStepR==50)
{
    CountTime();cap(h);
}
for(int x5=1;x5<=6000;x5++)
    _outpw(0x378,pr1);

//6
if(m_iStepR==200)
{
    CountTime();cap(h);
}
for(int x6=1;x6<=6000;x6++)
    _outpw(0x378,pr2);

//7
if(m_iStepR==200)
{
    CountTime();cap(h);
}
if(m_iStepR==100)
{
    CountTime();cap(h);
}
for(int x7=1;x7<=6000;x7++)
    _outpw(0x378,pr3);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

//8
if(m_iStepR==200)
{
    CountTime();cap(h);
}
for(int x8=1;x8<=6000;x8++)
    _outpw(0x378,pr4);
}
}

```

```

_outpw(0x378,0);
//rename("d:\photo\output.txt",m_sName); //เปลี่ยนชื่อfile ตามที่กรอกในinput dialog
WinExec("C:\\Program Files\\Microsoft Visual
Studio\\MyProjects\\display2\\Debug\\display2.exe",SW_SHOW);
//เปิดโปรแกรมแสดงผล

```

```

void CScanner1Dlg::OnTimer(UINT nIDEvent)
{
    // TODO: Add your message handler code here and/or call default
    //ส่วนแสดงเวลา
    CTime curTime =CTime::GetCurrentTime();

    if (firsttime==0)
    {
        oldsec=curTime.GetSecond();
        firsttime=1;
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if(oldsec>curTime.GetSecond())
    intersec=60+curTime.GetSecond()-oldsec;
else
    intersec=curTime.GetSecond()-oldsec;

oldsec=curTime.GetSecond();
usetime=usetime+intersec;
hour=int(usetime/3600);
minute=int(usetime/60);
second=int(usetime%60);

m_sTime.Format("%d:%d:%d",hour,minute,second);

CDialog::OnTimer(nIDEvent);
}

void CScanner1Dlg::CountTime()
{
    ,/นับเวลาdelay
    WaitTime=0;
    CTime curStartTime=CTime::GetCurrentTime();
    StartSec=curStartTime.GetSecond();

do
{
    CTime curTestTime=CTime::GetCurrentTime();
    TestSec=curTestTime.GetSecond();
    WaitTime=TestSec-StartSec;

    if(WaitTime<0)
    {
        MessageBeep((WORD)-1);
    }
}

```



```
WaitTime=WaitTime+60;
```

```
}
```

```
}
```

```
while(WaitTime<m_iDelayR);
```

```
void CScanner1Dlg::cap(int mh)
```

```
{
```

```
//FILE *capture;
```

```
//เข้ารหัสชื่อfile เพื่อที่จะได้รู้ว่าเป็นstep เท่าไร ชั้นที่เท่าไร
```

```
switch(mh)
```

```
{
```

```
case 1:PicName[10]='0';PicName[11]='1';next=1;break;
```

```
case 2:PicName[10]='0';PicName[11]='2';next=2;break;
```

```
case 3:PicName[10]='0';PicName[11]='3';next=3;break;
```

```
case 4:PicName[10]='0';PicName[11]='4';next=4;break;
```

```
case 5:PicName[10]='0';PicName[11]='5';next=5;break;
```

```
case 6:PicName[10]='0';PicName[11]='6';next=6;break;
```

```
case 7:PicName[10]='0';PicName[11]='7';next=7;break;
```

```
case 8:PicName[10]='0';PicName[11]='8';next=8;break;
```

```
case 9:PicName[10]='0';PicName[11]='9';next=9;break;
```

```
case 10:PicName[10]='1';PicName[11]='0';next=10;break;
```

```
case 11:PicName[10]='1';PicName[11]='1';next=11;break;
```

```
case 12:PicName[10]='1';PicName[11]='2';next=12;break;
```

```
case 13:PicName[10]='1';PicName[11]='3';next=13;break;
```

```
case 14:PicName[10]='1';PicName[11]='4';next=14;break;
```

```
case 15:PicName[10]='1';PicName[11]='5';next=15;break;
```

```
case 16:PicName[10]='1';PicName[11]='6';next=16;break;
```

```
case 17:PicName[10]='1';PicName[11]='7';next=17;break;
```

```
case 18:PicName[10]='1';PicName[11]='8';next=18;break;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

case 19:PicName[10]='1';PicName[11]='9';next=19;break;
case 20:PicName[10]='2';PicName[11]='0';next=20;break;

case 21:PicName[10]='2';PicName[11]='1';next=21;break;
case 22:PicName[10]='2';PicName[11]='2';next=22;break;
case 23:PicName[10]='2';PicName[11]='3';next=23;break;
case 24:PicName[10]='2';PicName[11]='4';next=24;break;
case 25:PicName[10]='2';PicName[11]='5';next=25;break;
case 26:PicName[10]='2';PicName[11]='6';next=26;break;
case 27:PicName[10]='2';PicName[11]='7';next=27;break;
case 28:PicName[10]='2';PicName[11]='8';next=28;break;
case 29:PicName[10]='2';PicName[11]='9';next=29;break;
case 30:PicName[10]='3';PicName[11]='0';next=30;break;
}

if(next!=OldNext)
{
    OldNext=OldNext+1;
    MessageBeep((WORD)-1);
    CountRound=0;
}
CountRound=CountRound+1;

switch(int(CountRound/100))
{
case 0:PicName[13]='0';break;
case 1:PicName[13]='1';break;
case 2:PicName[13]='2';break;
default: MessageBox("ERROR PicName[13]");
}

switch(int(((CountRound-(int(CountRound/100)*100))/10))

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{
case 0:PicName[14]='0';break;
case 1:PicName[14]='1';break;
case 2:PicName[14]='2';break;
case 3:PicName[14]='3';break;
case 4:PicName[14]='4';break;
case 5:PicName[14]='5';break;
case 6:PicName[14]='6';break;
case 7:PicName[14]='7';break;
case 8:PicName[14]='8';break;
case 9:PicName[14]='9';break;
}

```

```

switch (CountRound-(int)(CountRound/10)*10)
{
case 0:PicName[15]='0';break;
case 1:PicName[15]='1';break;
case 2:PicName[15]='2';break;
case 3:PicName[15]='3';break;
case 4:PicName[15]='4';break;
case 5:PicName[15]='5';break;
case 6:PicName[15]='6';break;
case 7:PicName[15]='7';break;
case 8:PicName[15]='8';break;
case 9:PicName[15]='9';break;
}

```

```
//Update SCREEN//
```

```

m_iNewRound=CountRound;
m_iTotalRound=m_iStepR;
m_iNewHight=mb;
m_iTotalHight=m_iAllhightR;
UpdateData(FALSE);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
RedrawWindow(NULL, NULL, RDW_INVALIDATE | RDW_UPDATENOW );
```

```
OnTimer(NULL);
```

```
//ส่วนการจับภาพ
```

```
//CapTure
```

```
CVidCap VidCap;
```

```
VidCap.Create(WS_CHILD, CRect(30,30, 100,300), this);
```

```
VidCap.DriverConnect(0);
```

```
//VidCap.AutoSize();
```

```
//VidCap.Preview(TRUE);
```

```
//VidCap.FileSaveDIB("D:\\photo\\save4.bmp");
```

```
//VidCap.CenterWindow();
```

```
VidCap.FileSaveDIB(PicName);
```

```
//Find Point in BMP
```

```
//หาจุดที่สว่างที่สุด
```

```
struct DATA
```

```
{
```

```
    unsigned char  abyte1;
```

```
    unsigned char  abyte2;
```

```
    unsigned char  abyte3;
```

```
}data;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

int r[300][30];
//int g[300][30];
//int b[300][30];
int d,dd,rmax,ry,pixelmaxh,pixelmaxw,lineredmax;
//
double rmaxx;
        double redmax;

```

```

FILE *bmp;
bmp=fopen(PicName,"rb+");
if (bmp == NULL)
{
    MessageBox("ERROR-Cannot open the designated file");
}
else
{
    redmax=0;
    lineredmax=0;
    dd=0;
    ry=0;
    pixelmaxh=0;
    pixelmaxw=0;
    for(unsigned int ssss=0;ssss<=299;ssss++)
    {
        if(ssss!=(299-41))
        {
            d=0;
            rmax=0;
            for(unsigned int sss=0;sss<=29;sss++)
            {
                fseek(bmp,54+(ssss*30*3)+(sss*3)+(ssss*2),SEEK_SET);
                fread(&data,sizeof(struct DATA),3,bmp);
                //b[299-ssss][sss]=data.abyte1;
            }
        }
    }
}

```

```

//g[299-ssss][sss]=data.abyte2;

r[299-ssss][sss]=data.abyte3;

//if(r[299-ssss][sss]>=205)
//{

//dd++;

//d++;

//cout<<" width :"<<sss<<" "<<"high :"<<(299-ssss)<<"
"<<r[299-ssss][sss]<<endl;

//    max=rmax+r[299-ssss][sss];
//}

if(r[299-ssss][sss]>ry)
{
    ry=r[299-ssss][sss];
    pixelmaxh=299-ssss;
    // pixelmaxw=sss;
}
}
}

//if(d>=1)
//{

//    maxx=rmax/d;
//    if(maxx>=redmax)
//    {
//        redmax=maxx;
//        linedredmax=299-ssss;
//        cout<<linedredmax<<" "<<redmax<<endl;
//    }

//}

//}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


```
void CScanner1Dlg::save(int pixelmaxh)
{
```

```
//เปลี่ยนข้อมูลจากpixel เป็น มิลลิเมตร
```

```
switch(int(pixelmaxh/100))
```

```
{
```

```
case 0: Hundred=0; break;
```

```
case 1: Hundred=1; break;
```

```
case 2: Hundred=2; break;
```

```
default :MessageBox("ERROR pixelmaxh OVER CASE");break;
```

```
}
```

```
if(Hundred==0)
```

```
{
```

```
switch(int(pixelmaxh/10))
```

```
{
```

```
case 0://000-009
```

```
switch(int(pixelmaxh%10))
```

```
{
```

```
case 0:meter=0;break;
```

```
case 1:meter=0;break;
```

```
case 2:meter=0;break;
```

```
case 3:meter=0;break;
```

```
case 4:meter=0;break;
```

```
case 5:meter=0;break;
```

```
case 6:meter=0;break;
```

```

case 6: meter=0; break;
case 7: meter=2; break;
case 8: meter=2; break;
case 9: meter=2; break;
}
break;
case 1://010-019
switch(int(pixelmaxh%10))
{
case 0: meter=2; break;
case 1: meter=4; break;
case 2: meter=4; break;
case 3: meter=4; break;
case 4: meter=4; break;
case 5: meter=6; break;
case 6: meter=6; break;
case 7: meter=6; break;
case 8: meter=6; break;
case 9: meter=8; break;
}
break;
case 2://020-029
switch(int(pixelmaxh%10))
{
case 0: meter=8; break;
case 1: meter=8; break;
case 2: meter=8; break;
case 3: meter=10; break;
case 4: meter=10; break;
case 5: meter=10; break;
case 6: meter=10; break;
case 7: meter=12; break;
case 8: meter=12; break;
case 9: meter=12; break;
}

```

```

        break;
    case 3://030-039
        switch(int(pixelmaxh%10))
        {
            case 0:meter=12;break;
            case 1:meter=14;break;
            case 2:meter=14;break;
            case 3:meter=14;break;
            case 4:meter=14;break;
            case 5:meter=16;break;
            case 6:meter=16;break;
            case 7:meter=16;break;
            case 8:meter=16;break;
            case 9:meter=18;break;
        }
        break;
    case 4://040-049
        switch(int(pixelmaxh%10))
        {
            case 0:meter=18;break;
            case 1:meter=18;break;
            case 2:meter=18;break;
            case 3:meter=20;break;
            case 4:meter=20;break;
            case 5:meter=20;break;
            case 6:meter=20;break;
            case 7:meter=22;break;
            case 8:meter=22;break;
            case 9:meter=22;break;
        }
        break;
    case 5://050-059
        switch(int(pixelmaxh%10))
        {
            case 0:meter=22;break;

```

```

case 1: meter=24; break;
case 2: meter=24; break;
case 3: meter=24; break;
case 4: meter=24; break;
case 5: meter=26; break;
case 6: meter=26; break;
case 7: meter=26; break;
case 8: meter=26; break;
case 9: meter=28; break;
}
break;
case 6://060-069
switch(int(pixelmaxh%10))
{
case 0: meter=28; break;
case 1: meter=28; break;
case 2: meter=28; break;
case 3: meter=30; break;
case 4: meter=30; break;
case 5: meter=30; break;
case 6: meter=30; break;
case 7: meter=32; break;
case 8: meter=32; break;
case 9: meter=32; break;
}
break;
case 7://070-079
switch(int(pixelmaxh%10))
{
case 0: meter=32; break;
case 1: meter=34; break;
case 2: meter=34; break;
case 3: meter=34; break;
case 4: meter=34; break;
case 5: meter=36; break;

```

```

        case 6: meter=36; break;
        case 7: meter=36; break;
        case 8: meter=36; break;
        case 9: meter=38; break;
    }
    break;
case 8://080-089
    switch(int(pixelmaxh%10))
    {
        case 0: meter=38; break;
        case 1: meter=38; break;
        case 2: meter=38; break;
        case 3: meter=40; break;
        case 4: meter=40; break;
        case 5: meter=40; break;
        case 6: meter=40; break;
        case 7: meter=42; break;
        case 8: meter=42; break;
        case 9: meter=42; break;
    }
    break;
case 9://090-099
    switch(int(pixelmaxh%10))
    {
        case 0: meter=42; break;
        case 1: meter=44; break;
        case 2: meter=44; break;
        case 3: meter=44; break;
        case 4: meter=44; break;
        case 5: meter=44; break;
        case 6: meter=46; break;
        case 7: meter=46; break;
        case 8: meter=46; break;
        case 9: meter=46; break;
    }
}

```

```

        break;
    }
}

if(Hundred==1)
{
    switch(int((pixelmaxh-100)/10))
    {
        case 0://100-109
            switch(int(pixelmaxh%10))
            {
                case 0:meter=46;break;
                case 1:meter=48;break;
                case 2:meter=48;break;
                case 3:meter=48;break;
                case 4:meter=48;break;
                case 5:meter=48;break;
                case 6:meter=50;break;
                case 7:meter=50;break;
                case 8:meter=50;break;
                case 9:meter=50;break;
            }
            break;
        case 1://110-119
            switch(int(pixelmaxh%10))
            {
                case 0:meter=50;break;
                case 1:meter=52;break;
                case 2:meter=52;break;
                case 3:meter=52;break;
                case 4:meter=52;break;
            }

```

```

case 5:meter=52;break;
case 6:meter=54;break;
case 7:meter=54;break;
case 8:meter=54;break;
case 9:meter=54;break;
}
break;
case 2://120-129
switch(int(pixelmaxh%10))
{
case 0:meter=54;break;
case 1:meter=56;break;
case 2:meter=56;break;
case 3:meter=56;break;
case 4:meter=56;break;
case 5:meter=56;break;
case 6:meter=58;break;
case 7:meter=58;break;
case 8:meter=58;break;
case 9:meter=58;break;
}
break;
case 3://130-139
switch(int(pixelmaxh%10))
{
case 0:meter=58;break;
case 1:meter=60;break;
case 2:meter=60;break;
case 3:meter=60;break;
case 4:meter=60;break;
case 5:meter=60;break;
case 6:meter=62;break;
case 7:meter=62;break;
case 8:meter=62;break;
case 9:meter=62;break;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    }
    break;
case 4://140-149
    switch(int(pixelmaxh%10))
    {
        case 0:meter=62;break;
        case 1:meter=64;break;
        case 2:meter=64;break;
        case 3:meter=64;break;
        case 4:meter=64;break;
        case 5:meter=64;break;
        case 6:meter=66;break;
        case 7:meter=66;break;
        case 8:meter=66;break;
        case 9:meter=66;break;
    }
    break;
case 5://150-159
    switch(int(pixelmaxh%10))
    {
        case 0:meter=66;break;
        case 1:meter=68;break;
        case 2:meter=68;break;
        case 3:meter=68;break;
        case 4:meter=68;break;
        case 5:meter=68;break;
        case 6:meter=70;break;
        case 7:meter=70;break;
        case 8:meter=70;break;
        case 9:meter=70;break;
    }
    break;
case 6://160-169
    switch(int(pixelmaxh%10))
    {

```

```

case 0:meter=70;break;
case 1:meter=72;break;
case 2:meter=72;break;
case 3:meter=72;break;
case 4:meter=72;break;
case 5:meter=72;break;
case 6:meter=74;break;
case 7:meter=74;break;
case 8:meter=74;break;
case 9:meter=74;break;
}
break;
case 7://170-179
switch(int(pixelmaxh%10))
{
case 0:meter=74;break;
case 1:meter=76;break;
case 2:meter=76;break;
case 3:meter=76;break;
case 4:meter=76;break;
case 5:meter=76;break;
case 6:meter=76;break;
case 7:meter=78;break;
case 8:meter=78;break;
case 9:meter=78;break;
}
break;
case 8://180-189
switch(int(pixelmaxh%10))
{
case 0:meter=78;break;
case 1:meter=78;break;
case 2:meter=78;break;
case 3:meter=80;break;
case 4:meter=80;break;

```

```

        case 5: meter=80; break;
        case 6: meter=80; break;
        case 7: meter=80; break;
        case 8: meter=80; break;
        case 9: meter=82; break;
    }
    break;
case 9://190-199
    switch(int(pixelmaxh%10))
    {
        case 0: meter=82; break;
        case 1: meter=82; break;
        case 2: meter=82; break;
        case 3: meter=82; break;
        case 4: meter=82; break;
        case 5: meter=82; break;
        case 6: meter=84; break;
        case 7: meter=84; break;
        case 8: meter=84; break;
        case 9: meter=84; break;
    }
    break;
}
}
if(Hundred==2)
{
    switch(int((pixelmaxh-200)/10))
    {
        case 0://200-209
            switch(int(pixelmaxh%10))
            {
                case 0: meter=84; break;
                case 1: meter=84; break;
                case 2: meter=84; break;

```

```

case 3:meter=86;break;
case 4:meter=86;break;
case 5:meter=86;break;
case 6:meter=86;break;
case 7:meter=86;break;
case 8:meter=86;break;
case 9:meter=86;break;
}

```

```
break;
```

```
case 1://210-219
```

```

switch(int(pixelmaxh%10))
{
case 0:meter=88;break;
case 1:meter=88;break;
case 2:meter=88;break;
case 3:meter=88;break;
case 4:meter=88;break;
case 5:meter=88;break;
case 6:meter=88;break;
case 7:meter=90;break;
case 8:meter=90;break;
case 9:meter=90;break;
}
break;

```

```
case 2://220-229
```

```

switch(int(pixelmaxh%10))
{
case 0:meter=90;break;
case 1:meter=90;break;
case 2:meter=90;break;
case 3:meter=90;break;
case 4:meter=92;break;
case 5:meter=92;break;
case 6:meter=92;break;
case 7:meter=92;break;
}

```

```

        case 8: meter=92; break;
        case 9: meter=92; break;
    }
    break;
case 3://230-239
    switch(int(pixelmaxh%10))
    {
        case 0: meter=92; break;
        case 1: meter=94; break;
        case 2: meter=94; break;
        case 3: meter=94; break;
        case 4: meter=94; break;
        case 5: meter=94; break;
        case 6: meter=94; break;
        case 7: meter=94; break;
        case 8: meter=96; break;
        case 9: meter=96; break;
    }
    break;
case 4://240-249
    switch(int(pixelmaxh%10))
    {
        case 0: meter=96; break;
        case 1: meter=96; break;
        case 2: meter=96; break;
        case 3: meter=96; break;
        case 4: meter=96; break;
        case 5: meter=96; break;
        case 6: meter=98; break;
        case 7: meter=98; break;
        case 8: meter=98; break;
        case 9: meter=98; break;
    }
    break;
case 5://250-259

```

```

switch(int(pixelmaxh%10))
{
case 0: meter=98; break;
case 1: meter=98; break;
case 2: meter=98; break;
case 3: meter=98; break;
case 4: meter=100; break;
case 5: meter=100; break;
case 6: meter=100; break;
case 7: meter=100; break;
case 8: meter=100; break;
case 9: meter=100; break;
}
break;
case 6://260-269
switch(int(pixelmaxh%10))
{
case 0: meter=100; break;
case 1: meter=100; break;
case 2: meter=102; break;
case 3: meter=102; break;
case 4: meter=102; break;
case 5: meter=102; break;
case 6: meter=102; break;
case 7: meter=102; break;
case 8: meter=102; break;
case 9: meter=102; break;
}
break;
case 7://270-279
switch(int(pixelmaxh%10))
{
case 0: meter=102; break;
case 1: meter=104; break;
case 2: meter=104; break;

```

```

case 3:meter=104;break;
case 4:meter=104;break;
case 5:meter=104;break;
case 6:meter=104;break;
case 7:meter=104;break;
case 8:meter=104;break;
case 9:meter=104;break;
}
break;

```

case 8://280-289

```

switch(int(pixelmaxh%10))
{
case 0:meter=106;break;
case 1:meter=106;break;
case 2:meter=106;break;
case 3:meter=106;break;
case 4:meter=106;break;
case 5:meter=106;break;
case 6:meter=106;break;
case 7:meter=106;break;
case 8:meter=106;break;
case 9:meter=108;break;
}
break;

```

case 9://290-299

```

switch(int(pixelmaxh%10))
{
case 0:meter=108;break;
case 1:meter=108;break;
case 2:meter=108;break;
case 3:meter=108;break;
case 4:meter=108;break;
case 5:meter=108;break;
case 6:meter=108;break;
case 7:meter=108;break;

```

```
case 9: meter=110; break;
```

```
}
```

```
break;
```

```
}
```

```
}
```

```
// ส่วนใส่ข้อมูลลงไปใน file
```

```
switch(int(meter/100))
```

```
{
```

```
case 0: OutRed100='0'; break;
```

```
case 1: OutRed100='1'; break;
```

```
case 2: OutRed100='2'; break;
```

```
case 3: OutRed100='3'; break;
```

```
case 4: OutRed100='4'; break;
```

```
case 5: OutRed100='5'; break;
```

```
case 6: OutRed100='6'; break;
```

```
case 7: OutRed100='7'; break;
```

```
case 8: OutRed100='8'; break;
```

```
case 9: OutRed100='9'; break;
```

```
}
```

```
switch(int((meter-(100*int(meter/100)))/10))
```

```
{
```

```
case 0: OutRed10='0'; break;
```

```
case 1: OutRed10='1'; break;
```

```
case 2: OutRed10='2'; break;
```

```
case 3: OutRed10='3'; break;
```

```
case 4: OutRed10='4'; break;
```

```
case 5: OutRed10='5'; break;
```

```
case 6: OutRed10='6'; break;
```

```

case 6:OutRed10='6';break;
case 7:OutRed10='7';break;
case 8:OutRed10='8';break;
case 9:OutRed10='9';break;
}
switch(int(meter%10))
{
case 0:OutRed1='0';break;
case 1:OutRed1='1';break;
case 2:OutRed1='2';break;
case 3:OutRed1='3';break;
case 4:OutRed1='4';break;
case 5:OutRed1='5';break;
case 6:OutRed1='6';break;
case 7:OutRed1='7';break;
case 8:OutRed1='8';break;
case 9:OutRed1='9';break;
}

```

```

- FILE *OutputWrite;
OutputWrite=fopen("D:\\photo\\output.txt","a");
fputc(OutRed10Q,OutputWrite);
fputc(OutRed10,OutputWrite);
fputc(OutRed1,OutputWrite);
//putc('X',OutputWrite);
putc(' ',OutputWrite);
fclose(OutputWrite);
}

```

```

void CScanner1Dlg::OnDel()
{

```

```
// TODO: Add your control notification handler code here
```

```
UpdateData(TRUE);
```

```
if(m_bDel==TRUE)
```

```
    delFactor=1;
```

```
else
```

```
    delFactor=0;
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DISPLAY โปรแกรมแสดงผล 3 มิติ จาก File display2Dlg.cpp

```

|
| \ display2Dlg.cpp : implementation file
|
|
|
|
| #include "stdafx.h" //ส่วน include header
| #include "display2.h"
| #include "display2Dlg.h"
| #include "PaintDlg.h"
| #include <math.h>
|
|
| #ifdef _DEBUG
| #define new DEBUG_NEW
| #undef THIS_FILE
| static char THIS_FILE[] = __FILE__;
| #endif
|
| double x[6001],y[6001],z[6001],r[6001],h[6001],Xd[6001],Yd[6001],X[6001],Y[6001],Z[6001];
|     int TotalH;
|     int OneRound;
|     double angle1;
|     double angleA=0;
|     double angleB=0;
|     double angleShift=3.14*2/360*10;
|     double angleX1=0;
|     double angleY1=0;
|     double angleZ1=0;
|     int CheckX,CheckY,CheckZ;
|     int CheckTX,CheckTY,CheckTZ;
|     char ch1,ch2;
|     int Tx,Ty,Tz;
|     int CheckColumn,CheckRow;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

// ส่วน about box

////////////////////////////////////

// CAboutDlg dialog used for App About

class CAboutDlg : public CDialog

public:

CAboutDlg();

// Dialog Data

//{{AFX_DATA(CAboutDlg)

enum { IDD = IDD_ABOUTBOX };

//}}AFX_DATA

// ClassWizard generated virtual function overrides

//{{AFX_VIRTUAL(CAboutDlg)

protected:

virtual void DoDataExchange(CDataExchange* pDX); // DDX/DDV support

//}}AFX_VIRTUAL

// Implementation

protected:

//{{AFX_MSG(CAboutDlg)

//}}AFX_MSG

DECLARE_MESSAGE_MAP()

};

CAboutDlg::CAboutDlg() : CDialog(CAboutDlg::IDD)

{

//{{AFX_DATA_INIT(CAboutDlg)

//}}AFX_DATA_INIT

}

void CAboutDlg::DoDataExchange(CDataExchange* pDX)

{

CDialog::DoDataExchange(pDX);

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
//{{AFX_DATA_MAP(CAboutDlg)
```

```
//}}AFX_DATA_MAP
```

```
BEGIN_MESSAGE_MAP(CAboutDlg, CDialog)
```

```
//{{AFX_MSG_MAP(CAboutDlg)
```

```
// No message handlers
```

```
//}}AFX_MSG_MAP
```

```
END_MESSAGE_MAP()
```

```
////////////////////////////////////
```

```
// CDisplay2Dlg dialog
```

```
CDisplay2Dlg::CDisplay2Dlg(CWnd* pParent /*=NULL*/)
:
```

```
CDialog(CDisplay2Dlg::IDD, pParent)
```

```
//{{AFX_DATA_INIT(CDisplay2Dlg)
```

```
m_iScX = 0.0;
```

```
m_iScY = 0.0;
```

```
m_iScZ = 0.0;
```

```
m_sOpenName = _T("");
```

```
m_bColumn = FALSE;
```

```
m_bRow = FALSE;
```

```
//}}AFX_DATA_INIT
```

```
// Note that LoadIcon does not require a subsequent DestroyIcon in Win32
```

```
m_hIcon = AfxGetApp()->LoadIcon(IDR_MAINFRAME);
```

```
void CDisplay2Dlg::DoDataExchange(CDataExchange* pDX)
```

```
{
```

```
CDialog::DoDataExchange(pDX);
```

```
//{{AFX_DATA_MAP(CDisplay2Dlg)
```

```
DDX_Control(pDX, IDC_BUTTON1, m_draw);
```

```
DDX_Text(pDX, IDC_SCX, m_iScX);
```

```
DDX_Text(pDX, IDC_SCY, m_iScY);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

DDX_Text(pDX, IDC_SCZ, m_iScZ);
DDX_Text(pDX, IDC_NAME, m_sOpenName);
DDX_Check(pDX, IDC_CHECK1, m_bColumn);
DDX_Check(pDX, IDC_CHECK2, m_bRow);
//}}AFX_DATA_MAP

```

```

BEGIN_MESSAGE_MAP(CDisplay2Dlg, CDialog)
//{{AFX_MSG_MAP(CDisplay2Dlg)
ON_WM_SYSCOMMAND()
ON_WM_PAINT()
ON_WM_QUERYDRAGICON()
ON_BN_CLICKED(IDC_BUTTON1, OnButton1)
ON_BN_CLICKED(IDC_XUP, OnXup)
ON_BN_CLICKED(IDC_XDOWN, OnXdown)
ON_BN_CLICKED(IDC_YUP, OnYup)
ON_BN_CLICKED(IDC_YDOWN, OnYdown)
ON_BN_CLICKED(IDC_ZDOWN, OnZdown)
ON_BN_CLICKED(IDC_ZUP, OnZup)
ON_BN_CLICKED(IDC_TXUP, OnTxup)
ON_BN_CLICKED(IDC_TXDOWN, OnTxdown)
ON_BN_CLICKED(IDC_TYUP, OnTyup)
ON_BN_CLICKED(IDC_TYDOWN, OnTydown)
ON_BN_CLICKED(IDC_TZUP, OnTzup)
ON_BN_CLICKED(IDC_TZDOWN, OnTzdown)
ON_BN_CLICKED(IDC_SC, OnSc)
ON_BN_CLICKED(IDC_FILE, OnFile)
ON_BN_CLICKED(IDC_CHECK1, OnCheck1)
ON_BN_CLICKED(IDC_CHECK2, OnCheck2)
ON_BN_CLICKED(IDC_BUTTON2, OnButton2)
//}}AFX_MSG_MAP
END_MESSAGE_MAP()

```

```

////////////////////////////////////

```

```

// CDisplay2Dlg message handlers

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
BOOL CDisplay2Dlg::OnInitDialog()
```

```
{
    CDialog::OnInitDialog();
```

```
    // Add "About..." menu item to system menu.
```

```
    // IDM_ABOUTBOX must be in the system command range.
```

```
    ASSERT((IDM_ABOUTBOX & 0xFFFF0) == IDM_ABOUTBOX);
```

```
    ASSERT(IDM_ABOUTBOX < 0xF000);
```

```
    CMenu* pSysMenu = GetSystemMenu(FALSE);
```

```
    if (pSysMenu != NULL)
```

```
{
```

```
        CString strAboutMenu;
```

```
        strAboutMenu.LoadString(IDS_ABOUTBOX);
```

```
        if (!strAboutMenu.IsEmpty())
```

```
{
```

```
            pSysMenu->AppendMenu(MF_SEPARATOR);
```

```
            pSysMenu->AppendMenu(MF_STRING, IDM_ABOUTBOX, strAboutMenu);
```

```
}
```

```
}
```

```
    // Set the icon for this dialog. The framework does this automatically
```

```
    // when the application's main window is not a dialog
```

```
    SetIcon(m_hIcon, TRUE);           // Set big icon
```

```
    SetIcon(m_hIcon, FALSE);        // Set small icon
```

```
    // TODO: Add extra initialization here
```

```
    // ส่วนประกาศค่าเริ่มต้น
```

```
    CheckColumn=1;
```

```
    CheckRow=1;
```

```
    m_bColumn=TRUE;
```

```
    m_bRow=TRUE;
```

```
    m_iScX=1;
```

```
    m_iScY=1;
```

```
m_iScZ=1;
```

```
//เปิด file เก็บข้อมูลแล้วเริ่มทำงาน
```

```
m_sOpenName="d:\\photo\\output.txt";
```

```
UpdateData(FALSE);
```

```
GetDlgItem(IDC_SCX) . ->EnableWindow(FALSE);
```

```
GetDlgItem(IDC_SCY) ->EnableWindow(FALSE);
```

```
GetDlgItem(IDC_SCZ) ->EnableWindow(FALSE);
```

```
GetDlgItem(IDC_SC) ->EnableWindow(FALSE);
```

```
GetDlgItem(IDC_XUP) ->EnableWindow(FALSE);
```

```
GetDlgItem(IDC_XDOWN) ->EnableWindow(FALSE);
```

```
GetDlgItem(IDC_YUP) ->EnableWindow(FALSE);
```

```
GetDlgItem(IDC_YDOWN) ->EnableWindow(FALSE);
```

```
GetDlgItem(IDC_ZUP) ->EnableWindow(FALSE);
```

```
GetDlgItem(IDC_ZDOWN) ->EnableWindow(FALSE);
```

```
GetDlgItem(IDC_TXUP) ->EnableWindow(FALSE);
```

```
GetDlgItem(IDC_TXDOWN) ->EnableWindow(FALSE);
```

```
GetDlgItem(IDC_TYUP) ->EnableWindow(FALSE);
```

```
GetDlgItem(IDC_TYDOWN) ->EnableWindow(FALSE);
```

```
GetDlgItem(IDC_TZUP) ->EnableWindow(FALSE);
```

```
GetDlgItem(IDC_TZDOWN) ->EnableWindow(FALSE);
```

```
GetDlgItem(IDC_STATICR) ->EnableWindow(FALSE);
```

```
GetDlgItem(IDC_STATICT) ->EnableWindow(FALSE);
```

```
GetDlgItem(IDC_STATICS) ->EnableWindow(FALSE);
```

```
GetDlgItem(IDC_STATICF) ->EnableWindow(FALSE);
```

```
GetDlgItem(IDC_STATICRX) ->EnableWindow(FALSE);
```

```
GetDlgItem(IDC_STATICRY) ->EnableWindow(FALSE);
```

```
GetDlgItem(IDC_STATICRZ) ->EnableWindow(FALSE);
```

```
GetDlgItem(IDC_STATICTX) ->EnableWindow(FALSE);
```

```
GetDlgItem(IDC_STATICTY) ->EnableWindow(FALSE);
```

```
GetDlgItem(IDC_STATICTZ) ->EnableWindow(FALSE);
```

```
GetDlgItem(IDC_STATICSX) ->EnableWindow(FALSE);
```

```
GetDlgItem(IDC_STATICSY) ->EnableWindow(FALSE);
```

```
GetDlgItem(IDC_STATICSZ) ->EnableWindow(FALSE);
```

```
GetDlgItem(IDC_CHECK1)        ->EnableWindow(FALSE);
```

```
GetDlgItem(IDC_CHECK2)        ->EnableWindow(FALSE);
```

```
return TRUE; // return TRUE unless you set the focus to a control
```

```
void CDisplay2Dlg::OnSysCommand(UINT nID, LPARAM lParam)
```

```
{
    if ((nID & 0xFFFF) == IDM_ABOUTBOX)
    {
        CAboutDlg dlgAbout;
        dlgAbout.DoModal();
    }
    else
    {
        CDialog::OnSysCommand(nID, lParam);
    }
}
```

// If you add a minimize button to your dialog, you will need the code below
 // to draw the icon. For MFC applications using the document/view model,
 // this is automatically done for you by the framework.

```
void CDisplay2Dlg::OnPaint()
```

```
{
    if (IsIconic())
    {
        CPaintDC dc(this); // device context for painting

        SendMessage(WM_ICONERASEBKGND, (LPARAM) dc.GetSafeHdc(), 0);

        // Center icon in client rectangle
        int cxIcon = GetSystemMetrics(SM_CXICON);
        int cyIcon = GetSystemMetrics(SM_CYICON);
        CRect rect;
```

```

GetClientRect(&rect);

int x = (rect.Width() - cxIcon + 1) / 2;
int y = (rect.Height() - cyIcon + 1) / 2;

```

```

// Draw the icon
dc.DrawIcon(x, y, m_hIcon);

```

```

}
else
{

```

```

    CDialog::OnPaint();
}
}

```

```

// The system calls this to obtain the cursor to display while the user drags
// the minimized window.

```

```

HCURSOR CDisplay2Dlg::OnQueryDragIcon()
{

```

```

    return (HCURSOR) m_hIcon;
}

```

```

//เมื่อคลิกปุ่มแสดงภาพ แสดงภาพ

```

```

void CDisplay2Dlg::OnButton1()
{

```

```

    // TODO: Add your control notification handler code here

```

```

GetDlgItem(IDC_SCX)         ->EnableWindow(TRUE);
GetDlgItem(IDC_SCY)         ->EnableWindow(TRUE);
GetDlgItem(IDC_SCZ)         ->EnableWindow(TRUE);
GetDlgItem(IDC_SC)          ->EnableWindow(TRUE);
GetDlgItem(IDC_XUP)         ->EnableWindow(TRUE);
GetDlgItem(IDC_XDOWN)       ->EnableWindow(TRUE);
GetDlgItem(IDC_YUP)         ->EnableWindow(TRUE);
GetDlgItem(IDC_YDOWN)       ->EnableWindow(TRUE);
GetDlgItem(IDC_ZUP)         ->EnableWindow(TRUE);
GetDlgItem(IDC_ZDOWN)       ->EnableWindow(TRUE);
GetDlgItem(IDC_TXUP)        ->EnableWindow(TRUE);

```

```

GetDlgItem(IDC_TXDOWN)          ->EnableWindow(TRUE);
GetDlgItem(IDC_TYUP)           ->EnableWindow(TRUE);
GetDlgItem(IDC_TYDOWN)        ->EnableWindow(TRUE);
GetDlgItem(IDC_TZUP)          ->EnableWindow(TRUE);
GetDlgItem(IDC_TZDOWN)        ->EnableWindow(TRUE);
GetDlgItem(IDC_STATICR)       ->EnableWindow(TRUE);
GetDlgItem(IDC_STATICT)       ->EnableWindow(TRUE);
GetDlgItem(IDC_STATICS)       ->EnableWindow(TRUE);
GetDlgItem(IDC_STATICF)       ->EnableWindow(TRUE);
GetDlgItem(IDC_STATICRX)      ->EnableWindow(TRUE);
GetDlgItem(IDC_STATICRY)      ->EnableWindow(TRUE);
GetDlgItem(IDC_STATICRZ)      ->EnableWindow(TRUE);
GetDlgItem(IDC_STATICTX)      ->EnableWindow(TRUE);
GetDlgItem(IDC_STATICTY)      ->EnableWindow(TRUE);
GetDlgItem(IDC_STATICTZ)      ->EnableWindow(TRUE);
GetDlgItem(IDC_STATICSX)      ->EnableWindow(TRUE);
GetDlgItem(IDC_STATICSY)      ->EnableWindow(TRUE);
GetDlgItem(IDC_STATICSZ)      ->EnableWindow(TRUE);
GetDlgItem(IDC_CHECK1)        ->EnableWindow(TRUE);
GetDlgItem(IDC_CHECK2)        ->EnableWindow(TRUE);

FILE *fp;
    if((fp=fopen(m_sOpenName,"r+"))==NULL)
    {
        MessageBox("CANNOT OPEN D:\\PHOTO\\OUTPUT.TXT");
    }
char OneRound1=getc(fp);
char OneRound2=getc(fp);
char OneRound3=getc(fp);
OneRound=(100*(OneRound1-48))+(10*(OneRound2-48))+(OneRound3-48);

char temp1=getc(fp);
if(temp1!=' ')
{

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    MessageBox("ERROR WITH TEMP1");
}

angle1=3.14*2/OneRound;

char TotalH1=getc(fp);
char TotalH2=getc(fp);
TotalH=OneRound*((10*(TotalH1-48))+(TotalH2-48));

char temp2=getc(fp);
if(temp2!=' ')
{
    MessageBox("ERROR WITH TEMP2");
}

char HightInCm1=getc(fp);
char HightInCm2=getc(fp);
int HightInCm=(10*(HightInCm1-48))+(HightInCm2-48);

char temp3=getc(fp);
if(temp3!=' ')
{
    MessageBox("ERROR WITH TEMP3");
}

for(int u=1;u<=TotalH;u++)
{
    char r1=getc(fp);
    char r2=getc(fp);
    char r3=getc(fp);
    r[u]=(100*(r1-48))+(10*(r2-48))+(r3-48);

    char tempDATA=getc(fp);

```

```

if(tempDATA!=' ')
{
    MessageBox("ERROR WITH TEMPDATA");
}
}

```

```
fclose(fp);
```

```
double r[101]=
```

```
{
```

```
50,
```

```
52, 54, 54, 56, 58, 62, 66, 56, 52, 48,
```

```
42, 42, 38, 38, 38, 44, 42, 46, 52, 56,
```

```
54, 48, 44, 42, 38, 38, 38, 40, 44, 46,
```

```
48, 52, 58, 68, 64, 56, 56, 54, 54, 56,
```

```
54, 56, 62, 66, 72, 72, 62, 60, 56, 54,
```

```
54, 54, 54, 56, 58, 62, 68, 58, 50, 48,
```

```
44, 40, 42, 38, 38, 40, 44, 46, 50, 56,
```

```
54, 50, 44, 42, 40, 42, 38, 44, 44, 46,
```

```
50, 52, 60, 68, 64, 60, 56, 56, 52, 56,
```

```
54, 56, 60, 68, 72, 74, 64, 58, 56, 56
```

```
};
```

```
for (int i1=1;i1<=TotalH;i1++)
```

```
{
```

```
x[i1]=1.5*r[i1]*cos(angle1*(i1%TotalH));
```

```
y[i1]=1.5*r[i1]*sin(angle1*(i1%TotalH));
```

```
z[i1]=HightInCm*1.35*int((i1-1)/OneRound);
```

```
X[i1]=x[i1];
```

```
Y[i1]=y[i1];
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Z[i1]=z[i1];
```

```
}
```

```
RedrawWindow(NULL, NULL, RDW_INVALIDATE | RDW_UPDATENOW | RDW_ERASE );
```

```
angleX1=0;
```

```
angleY1=0;
```

```
angleZ1=0;
```

```
Tx=0;
```

```
Ty=0;
```

```
Tz=0;
```

```
draw(OneRound,TotalH);
```

```
//การ rotation
```

```
void CDisplay2Dlg::OnXup()
```

```
{
```

```
// TODO: Add your control notification handler code here
```

```
if(CheckX==1)
```

```
{
```

```
    //MessageBox("inX");
```

```
    angleX1=0;
```

```
    for (int i5=1;i5<=TotalH;i5++)
```

```
    {
```

```
        x[i5]=X[i5];
```

```
        y[i5]=Y[i5];
```

```
        z[i5]=Z[i5];
```

```
    }
```

```
}
```

```
angleX1=angleX1+angleShift;
```

```
for (int i4=1;i4<=TotalH;i4++)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{
    X[i4]=x[i4];
    Y[i4]=y[i4]*cos(angleX1)-z[i4]*sin(angleX1);
    Z[i4]=y[i4]*sin(angleX1)+z[i4]*cos(angleX1);
}

RedrawWindow(NULL, NULL, RDW_INVALIDATE | RDW_UPDATENOW | RDW_ERASE );
draw(OneRound,TotalH);
CheckX=0;CheckY=1;CheckZ=1;CheckTX=1;CheckTY=1;CheckTZ=1;
}
}

```

```

void CDisplay2Dlg::OnXdown()
{

```

```

// TODO: Add your control notification handler code here

```

```

if(CheckX==1)
{

```

```

    angleX1=0;

```

```

    //MessageBox("inX");

```

```

    for (int i5=1;i5<=TotalH;i5++)
    {

```

```

        x[i5]=X[i5];

```

```

        y[i5]=Y[i5];

```

```

        z[i5]=Z[i5];
    }
}

```

```

}

```

```

angleX1=angleX1-angleShift;

```

```

for (int i4=1;i4<=TotalH;i4++)

```

```

{
    X[i4]=x[i4];
    Y[i4]=y[i4]*cos(angleX1)-z[i4]*sin(angleX1);
    Z[i4]=y[i4]*sin(angleX1)+z[i4]*cos(angleX1);
}

RedrawWindow(NULL, NULL, RDW_INVALIDATE | RDW_UPDATENOW | RDW_ERASE );
draw(OneRound,TotalH);
CheckX=0;CheckY=1;CheckZ=1;CheckTX=1;CheckTY=1;CheckTZ=1;
}

```

```
void CDisplay2Dlg::OnYup()
```

```
// TODO: Add your control notification handler code here
```

```
if(CheckY==1)
```

```
{
```

```
    angleY1=0;
```

```
    //MessageBox("inY");
```

```
    for (int i5=1;i5<=TotalH;i5++)
```

```
    {
```

```
        x[i5]=X[i5];
```

```
        y[i5]=Y[i5];
```

```
        z[i5]=Z[i5];
```

```
    }
```

```
}
```

```
angleY1=angleY1+angleShift;
```

```
for (int i4=1;i4<=TotalH;i4++)
```

```
{
```

```
    X[i4]=x[i4]*cos(angleY1)+z[i4]*sin(angleY1);
```

```
    Y[i4]=y[i4];
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Z[i4]=-1*x[i4]*sin(angleY1)+z[i4]*cos(angleY1);
```

```
}
```

```
RedrawWindow(NULL, NULL, RDW_INVALIDATE | RDW_UPDATENOW | RDW_ERASE );
```

```
draw(OneRound,TotalH);
```

```
CheckX=1;CheckY=0;CheckZ=1;CheckTX=1;CheckTY=1;CheckTZ=1;
```

```
}
```

```
void CDisplay2Dlg::OnYdown()
```

```
{
```

```
// TODO: Add your control notification handler code here
```

```
if(CheckY==1)
```

```
{
```

```
angleY1=0;
```

```
//MessageBox("inY");
```

```
for (int i5=1;i5<=TotalH;i5++)
```

```
{
```

```
x[i5]=X[i5];
```

```
y[i5]=Y[i5];
```

```
z[i5]=Z[i5];
```

```
}
```

```
}
```

```
angleY1=angleY1-angleShift;
```

```
for (int i4=1;i4<=TotalH;i4++)
```

```
{
```

```
X[i4]=x[i4]*cos(angleY1)+z[i4]*sin(angleY1);
```

```
Y[i4]=y[i4];
```

```
Z[i4]=-1*x[i4]*sin(angleY1)+z[i4]*cos(angleY1);
```

```
}
```

```
RedrawWindow(NULL, NULL, RDW_INVALIDATE | RDW_UPDATENOW | RDW_ERASE );
```

```
draw(OneRound,TotalH);
```

```
CheckX=1;CheckY=0;CheckZ=1;CheckTX=1;CheckTY=1;CheckTZ=1;
```

```
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

void CDisplay2Dlg::OnZup()
{
    // TODO: Add your control notification handler code here

    if(CheckZ==1)
    {
        //MessageBox("inZ");
        angleZ1=0;
        for (int i5=1;i5<=TotalH;i5++)
        {
            x[i5]=X[i5];
            y[i5]=Y[i5];
            z[i5]=Z[i5];
        }
    }

    angleZ1=angleZ1+angleShift;
    for (int i4=1;i4<=TotalH;i4++)
    {
        X[i4]=x[i4]*cos(angleZ1)-y[i4]*sin(angleZ1);
        Y[i4]=x[i4]*sin(angleZ1)+y[i4]*cos(angleZ1);
        Z[i4]=z[i4];
    }

    RedrawWindow(NULL, NULL, RDW_INVALIDATE | RDW_UPDATENOW | RDW_ERASE );
    draw(OneRound,TotalH);
    CheckX=1;CheckY=1;CheckZ=0;CheckTX=1;CheckTY=1;CheckTZ=1;
}

void CDisplay2Dlg::OnZdown()
{
    // TODO: Add your control notification handler code here

```

```

if(CheckZ==1)
{
    //MessageBox("inZ");
    angleZ1=0;
    for (int i5=1;i5<=TotalH;i5++)
    {
        x[i5]=X[i5];
        y[i5]=Y[i5];
        z[i5]=Z[i5];
    }
}

angleZ1=angleZ1-angleShift;
for (int i4=1;i4<=TotalH;i4++)
{
    X[i4]=x[i4]*cos(angleZ1)-y[i4]*sin(angleZ1);
    Y[i4]=x[i4]*sin(angleZ1)+y[i4]*cos(angleZ1);
    Z[i4]=z[i4];
}

RedrawWindow(NULL, NULL, RDW_INVALIDATE | RDW_UPDATENOW | RDW_ERASE );
draw(OneRound,TotalH);
CheckX=1;CheckY=1;CheckZ=0;CheckTX=1;CheckTY=1;CheckTZ=1;
}

```

```

void CDisplay2Dfg::draw(int OneRound,int TotalH)
{

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

for(int i2=1;i2<=TotalH;i2++)
{
    Xd[i2]=X[i2]*cos(angleA)-Y[i2]*sin(angleA);
    Yd[i2]=X[i2]*sin(angleA)*sin(angleB)+Y[i2]*cos(angleA)*sin(angleB)+Z[i2]*cos
(angleB);
}

CClientDC draw(this);
//draw.SetPixel(375,250,RGB(0,0,0));//319 239
//for(int i3=1;i3<=TotalH;i3++)
//{
//draw.SetPixel((319+int(Xd[i3])),(239+int(Yd[i3])),RGB(0,0,0));
//}
//CDC dc(this);
CClientDC dc(this);
CPen IPen(PS_SOLID,1,RGB(0,0,255));
CPen mPen(PS_SOLID,1,RGB(0,255,255));
CPen oPen(PS_SOLID,1,RGB(255,255,255));
dc.SelectObject(&oPen);
dc.MoveTo(375,247);
dc.LineTo(375,253);
dc.MoveTo(372,250);
dc.LineTo(378,250);

if(CheckColumn==1)
{
    for(int j1=0;j1<=int(TotalH/OneRound);j1++)
    {
        for(int i3=1;i3<=OneRound-1;i3++)
        {
            dc.SelectObject(&IPen);
            dc.MoveTo((375+int(Xd[j1*OneRound+i3]+0.5)),(250+int(Yd
[j1*OneRound+i3]+0.5)));

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


```

//MessageBox("inX");
Tx=0;

for (int i5=1;i5<=TotalH;i5++)
{
    x[i5]=X[i5];
    y[i5]=Y[i5];
    z[i5]=Z[i5];
}

```

```
Tx=Tx+10;
```

```
for (int i4=1;i4<=TotalH;i4++)
```

```

{
    X[i4]=x[i4]+Tx;
    Y[i4]=y[i4];
    Z[i4]=z[i4];
}

```

```
RedrawWindow(NULL, NULL, RDW_INVALIDATE | RDW_UPDATENOW | RDW_ERASE );
```

```
draw(OneRound,TotalH);
```

```
CheckX=1;CheckY=1;CheckZ=1;CheckTX=0;CheckTY=1;CheckTZ=1;
```

```
void CDisplay2Dlg::OnTxdown()
```

```

{
    // TODO: Add your control notification handler code here

```

```
if(CheckTX==1)
```

```

{
    //MessageBox("inX");

```

```
Tx=0;
```

```
for (int i5=1;i5<=TotalH;i5++)
```

```

{
    x[i5]=X[i5];

```

```
y[i5]=Y[i5];
```

```
z[i5]=Z[i5];
```

```
}
```

```
}
```

```
Tx=Tx-10;
```

```
for (int i4=1;i4<=TotalH;i4++)
```

```
{
```

```
    X[i4]=x[i4]+Tx;
```

```
    Y[i4]=y[i4];
```

```
    Z[i4]=z[i4];
```

```
}
```

```
RedrawWindow(NULL, NULL, RDW_INVALIDATE | RDW_UPDATENOW | RDW_ERASE );
```

```
draw(OneRound,TotalH);
```

```
CheckX=1;CheckY=1;CheckZ=1;CheckTX=0;CheckTY=1;CheckTZ=1;
```

```
void CDisplay2Dlg::OnTyup()
```

```
{
```

```
    // TODO: Add your control notification handler code here
```

```
    if(CheckTY==1)
```

```
{
```

```
    //MessageBox("inX");
```

```
    Ty=0;
```

```
    for (int i5=1;i5<=TotalH;i5++)
```

```
{
```

```
        x[i5]=X[i5];
```

```
        y[i5]=Y[i5];
```

```
        z[i5]=Z[i5];
```

```
}
```

```
}
```

```
Ty=Ty+10;
```

```
for (int i4=1;i4<=TotalH;i4++)
```

```
{
```

```
    X[i4]=x[i4];
```

```
    Y[i4]=y[i4]+Ty;
```

```
    Z[i4]=z[i4];
```

```
}
```

```
RedrawWindow(NULL, NULL, RDW_INVALIDATE | RDW_UPDATENOW | RDW_ERASE );
```

```
draw(OneRound,TotalH);
```

```
CheckX=1;CheckY=1;CheckZ=1;CheckTX=1;CheckTY=0;CheckTZ=1;
```

```
}
```

```
void CDisplay2Dlg::OnTydown()
```

```
{
```

```
    // TODO: Add your control notification handler code here
```

```
    if(CheckTY==1)
```

```
{
```

```
    //MessageBox("inX");
```

```
    Ty=0;
```

```
    for (int i5=1;i5<=TotalH;i5++)
```

```
    {
```

```
        x[i5]=X[i5];
```

```
        y[i5]=Y[i5];
```

```
        z[i5]=Z[i5];
```

```
    }
```

```
}
```

```
Ty=Ty-10;
```

```
for (int i4=1;i4<=TotalH;i4++)
```

```
{
```

```
    X[i4]=x[i4];
```

```
    Y[i4]=y[i4]+Ty;
```

```
    Z[i4]=z[i4];
```

```
}
```

```

RedrawWindow(NULL, NULL, RDW_INVALIDATE | RDW_UPDATENOW | RDW_ERASE );
draw(OneRound,TotalH);
CheckX=1;CheckY=1;CheckZ=1;CheckTX=1;CheckTY=0;CheckTZ=1;

```

```

void CDisplay2Dlg::OnTzup()

```

```

// TODO: Add your control notification handler code here

```

```

    if(CheckTZ==1)

```

```

    {

```

```

        //MessageBox("inX");

```

```

        Tz=0;

```

```

        for (int i5=1;i5<=TotalH;i5++)

```

```

        {

```

```

            x[i5]=X[i5];

```

```

            y[i5]=Y[i5];

```

```

            z[i5]=Z[i5];

```

```

        }

```

```

    }

```

```

Tz=Tz+10;

```

```

for (int i4=1;i4<=TotalH;i4++)

```

```

{

```

```

    X[i4]=x[i4];

```

```

    Y[i4]=y[i4];

```

```

    Z[i4]=z[i4]+Tz;

```

```

}

```

```

RedrawWindow(NULL, NULL, RDW_INVALIDATE | RDW_UPDATENOW | RDW_ERASE );

```

```

draw(OneRound,TotalH);

```

```

CheckX=1;CheckY=1;CheckZ=1;CheckTX=1;CheckTY=1;CheckTZ=0;

```

```

}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

void CDisplay2Dlg::OnTzdown()
{
    // TODO: Add your control notification handler code here
    if(CheckTZ==1)
    {
        //MessageBox("inX");
        Tz=0;

        for (int i5=1;i5<=TotalH;i5++)
        {
            x[i5]=X[i5];
            y[i5]=Y[i5];
            z[i5]=Z[i5];
        }
    }

    Tz=Tz-10;
    for (int i4=1;i4<=TotalH;i4++)
    {
        X[i4]=x[i4];
        Y[i4]=y[i4];
        Z[i4]=z[i4]+Tz;
    }

    RedrawWindow(NULL, NULL, RDW_INVALIDATE | RDW_UPDATENOW | RDW_ERASE );
    draw(OneRound,TotalH);

    CheckX=1;CheckY=1;CheckZ=1;CheckTX=1;CheckTY=1;CheckTZ=0;
}

```

//การ Scaling

```

void CDisplay2Dlg::OnSc()
{
    // TODO: Add your control notification handler code here
    UpdateData(TRUE);
    for (int i5=1;i5<=TotalH;i5++)

```

```

        {
            x[i5]=X[i5];
            y[i5]=Y[i5];
            z[i5]=Z[i5];
        }
for (int i4=1;i4<=TotalH;i4++)
{
    X[i4]=x[i4]*(m_iScX);
    Y[i4]=y[i4]*(m_iScY);
    Z[i4]=z[i4]*(m_iScZ);
}

RedrawWindow(NULL, NULL, RDW_INVALIDATE | RDW_UPDATENOW | RDW_ERASE );
draw(OneRound,TotalH);
m_iScX=1;
m_iScY=1;
m_iScZ=1;
UpdateData(FALSE);
CheckX=1;CheckY=1;CheckZ=1;CheckTX=1;CheckTY=1;CheckTZ=1;
}

```

// function เปิด file ที่ต้องการ
void CDisplay2Dlg::OnFile()

```

{
    // TODO: Add your control notification handler code here

    CFileDialog OpenName(TRUE,"111","d:\\3ddata\\output.txt",OFN_HIDEREADONLY |
OFN_OVERWRITEPROMPT,
        "TXT File (*.txt)|*.txt|3Ddata File (*.3dd)|*.3dd|All Files (*.*)|*.*|");
    if(OpenName.DoModal()==IDOK)

        m_sOpenName=OpenName.GetPathName();
    UpdateData(FALSE);
    OnButton1();
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void CDisplay2Dlg::OnCheck1()
```

```
{
// TODO: Add your control notification handler code here
UpdateData(TRUE);
if(m_bColumn==TRUE)
    CheckColumn=1;
else
    CheckColumn=0;
OnButton1();
}
```

```
void CDisplay2Dlg::OnCheck2()
```

```
{
// TODO: Add your control notification handler code here
UpdateData(TRUE);
if(m_bRow==TRUE)
    CheckRow=1;
else
    CheckRow=0;
OnButton1();
}
```

```
void CDisplay2Dlg::OnButton2()
```

```
{
// TODO: Add your control notification handler code here
OnCancel();
}
```

