สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ระบบสแกนพื้นผิวโดยเลเซอร์ 2

นายรุ่งเรือง แช่โค้ว
นายสมศักดิ์ ปัณฑุวงศ์

ปพ.
ร635ร
2536
เลขหมู่.....
เลขทะเบียน.....
วัน.เดือน.ปี.....

L12055046

โครงงานนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตร์บัณฑิต
ภาควิชาฟิสิกส์ประยุกต์
คณะวิทยาศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2536

Laser Surface Scanning System 2

Mr.Rungrueng Saekow

Mr.Somsak Puntuwong

A Special Project Submitted in Partial Fulfillment of the

Requirement for the Degree of Bachelor of Science

Department of Applied Physics

Faculty of Science

King Mongkut's Institute of Technology Ladkrabang

1993

หัวข้อโครงงานพิเศษ	ระบบสแกนพื้นผิวโดยเลเซอร์ 2
โดย	นายรุ่งเรือง แซ่โค้ว
	นายสมศักดิ์ ปัณฑวงศ์
ภาควิชา	ฟิสิกส์ประยุกต์
อาจารย์ที่ปรึกษา	ผศ.ดร.วราวุฒิ เถาลัดดา
	ผศ.ดร.อารีย์ วิเชียรฉาย

ภาควิชาฟิสิกส์ประยุกต์ คณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบังอนุมัติให้นับโครงงานพิเศษฉบับนี้เป็นส่วนหนึ่งของการศึกษา ตามหลักสูตรวิทยาศาสตร์บัณฑิต

..... หัวหน้าภาควิชาฟิสิกส์ประยุกต์

(ผศ. ปรีชา เทียนสมประสงค์)

คณะกรรมการสอบโครงงานพิเศษ

..... ประธานกรรมการ

(ผศ.ดร.วราวุฒิ เถาลัดดา)

..... กรรมการ

(ผศ.ดร.อารีย์ วิเชียรฉาย)

..... กรรมการ

(ผศ.ดร.ปรีชา เทียนสมประสงค์)

..... กรรมการ

(ผศ.เครือวัลย์ ศีตะจิตต์)

ลิขสิทธิ์ของภาควิชาฟิสิกส์ประยุกต์ คณะวิทยาศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

หัวข้อโครงงานพิเศษ	ระบบสแกนพื้นผิวโดยเลเซอร์ 2
โดย	นายรุ่งเรือง แซ่โค้ว
	นายสมศักดิ์ ปัณฑุวงศ์
ภาควิชา	ฟิสิกส์ประยุกต์
อาจารย์ที่ปรึกษา	ผศ.ดร.วราวุฒิ เกาลัดดา
	ผศ.ดร.อารีย์ วิเชียรฉาย
ปีการศึกษา	2536

บทคัดย่อ

โครงงานพิเศษนี้เป็นการสร้างภาพจากการเก็บข้อมูลจากความตื้นลึกของพื้นผิววัตถุโดยใช้ Laser Displacement Sensor (LDS) ทำการสแกนเก็บข้อมูลในแนวแกน X และ Y ของวัตถุที่จะทำการสร้างภาพบน XY-Table ซึ่งควบคุมการเคลื่อนที่ของแท่นในระนาบ XY โดยใช้สเตปปิ้งมอเตอร์ 2 ตัวเป็นตัวควบคุมสเตปปิ้งมอเตอร์ถูกควบคุมโดยไมโครคอมพิวเตอร์โดยผ่านการ์ด 8255 ผ่านวงจรขับสเตปปิ้งมอเตอร์ (Stepping Motor Driver Circuit) และทำการเก็บข้อมูลที่ได้จาก Laser Displacement Sensor ผ่านวงจรแปลงสัญญาณอนาลอกเป็นดิจิตอล (A/D Convertor) เก็บข้อมูลที่ได้เข้าไมโครคอมพิวเตอร์เพื่อสร้างภาพในลักษณะต่าง ๆโดยใช้ภาษาปาสคาลทำประมวลผลสร้างภาพ ตลอดจนควบคุมการสแกนเก็บข้อมูล ซึ่งสามารถแสดงผลในลักษณะภาพจำลองสามมิติได้ความละเอียดและความชัดเจนเป็นที่น่าพอใจ

Special Project Title	Laser Surface Scanning System 2
Name	Mr.Rungrueng Saekow
	Mr.Somsak Puntuwong
Special Project Advisor	Dr.Warawoot Thowladda
	Dr.Aree Wichaeanchai
Department	Applied Physics
Acedemic Year	1993

Abstract

Laser Surface Scanning System developed in this special project has an objective to perform an image of roughness surface. The thickness of sample surface which placed on the X-Y scan table is scanned along X and Y axis by using two stepping motors. Stepping motors are controlled through IC8255PPI by microcomputer. Analog thickness data are converted to digital data by 12 bit ADC. The image of scanned surface can be display in 3-D graphic.

กิตติกรรมประกาศ

โครงงานพิเศษนี้สำเร็จได้ด้วยดีด้วยความร่วมมือและความช่วยเหลือสนับสนุนจากบุคคลต่าง ๆ ซึ่งต้องขอขอบคุณบุคคลดังต่อไปนี้

บิดา-มารดาตลอดจนผู้มีอุปการะคุณทุกท่านที่ได้ให้กำเนิด เลี้ยงดู ให้การศึกษาและช่วยเหลือในทุก ๆ ด้าน

ผศ.ดร.วราวุฒิ เถาลัดดา ที่คอยให้ความช่วยเหลือและให้คำปรึกษาแนะนำเป็นแนวทางและติดตามผลจนโครงงานพิเศษนี้สำเร็จลุล่วงไปด้วยดี

ผศ.ดร.อารีย์ วิเชียรฉาย ที่ช่วยเหลือในด้านการจัดหาหัวข้อโครงงานพิเศษ ตลอดจนช่วยจัดหา Laser Displacement Sensor

อ.วิชิต ศิริโชติ ที่คอยให้คำปรึกษาและให้คำแนะนำตลอดจนคอยถามไถ่ถึงความคืบหน้าของโครงงานพิเศษ

คุณพิศาล งามเมืองแมน ,คุณสราวุฒิ วณิชชานุกร ที่ให้คำปรึกษาและแก้ปัญหาในด้านต่าง ๆ

คุณสาธิต ทองจีน ,คุณทรงวิทย์ เจริญรวยวัฒนา ที่คอยให้คำปรึกษาและแก้ปัญหาทางด้าน Software มาโดยตลอด

คุณกิตติชัย อภินทนาพงศ์ ,คุณชาญวิทย์ เดชอัศวนง ,คุณภูเบศร์ อุดมทรัพย์ และคุณสุภัทชดา ผาสุข ที่ช่วยออกความคิดให้คำปรึกษา และแก้ปัญหาต่าง ๆ ที่เกิดในโครงงานมาโดยตลอด

คุณบุญชัย พจมานสมสมาน ,คุณสมนึก วิเศษพานิชกิจ ที่เอื้อเฟื้อสถานที่ในการทำงาน

คุณกิตติชัย พิศาลวงศ์วาน ,คุณนรสีห์ สิงหลกะ ที่เอื้อเฟื้อช่วยถ่ายภาพ

ท้ายที่สุดขอขอบคุณพี่ ๆ เพื่อน ๆ และน้อง ๆ ทุกคนที่ให้ความสนใจคอยไต่ถามและให้กำลังใจจนโครงงานพิเศษนี้สำเร็จลุล่วงไปด้วยดี

รุ่งเรือง แซ่โค้ว

สมศักดิ์ ปัณฑุวงศ์

สารบัญ

หน้า

บทคัดย่อภาษาไทย ..... ก

บทคัดย่อภาษาอังกฤษ ..... ข

กิตติกรรมประกาศ ..... ค

สารบัญรูป ..... ง

บทที่ 1 บทนำ 1

บทที่ 2 การตรวจสอบพื้นผิววัตถุ 4

การศึกษาคุณสมบัติของพื้นผิววัตถุ 4

ตัวตรวจวัดระยะชัดด้วยเลเซอร์ 5

ระบบเลนส์ที่ใช้ 7

ตัวตรวจจับตำแหน่ง 8

- PSD (Position Sensitive Device) 8

- ระบบการประมวลสัญญาณ 9

คุณสมบัติของเลเซอร์ดิสเพลซเมนต์เซ็นเซอร์ 10

- ช่วงระยะการวัด 10

- สัญญาณเอาท์พุทกับความเร็วของการวัด 11

- ค่าความละเอียดในการวัด 11

บทที่ 3 ระบบขับเคลื่อนวัตถุ 13

สเตปปิ้งมอเตอร์ 14

- ระบบสเตปปิ้งมอเตอร 14

- แรงบิด 15

ระบบขับเคลื่อน 16

ระบบควบคุม 17

- หลักการขับมอเตอร์แบบกระตุ้นเฟสเดียว 17

- หลักการขับมอเตอร์แบบกระตุ้นหนึ่งและสองเฟส 17

- การแกว่งเข้าสู่สภาวะคงตัว 18

บทที่ 4 การเชื่อมต่อกับไมโครคอมพิวเตอร์ไอบีเอ็มพีซี 20

การเชื่อมต่อไมโครคอมพิวเตอร์ 20
- การแบ่งแอดเดรสของ IBM/PC 21
วงจรรับส่งข้อมูลแบบขนาน 22
- การ Decode address 22
วงจรแปลงสัญญาณอนาลอกเป็นดิจิตอล 25
- A/D แบบประมาณค่าผลสำเร็จ 25
- คุณสมบัติของ ADC574AJ กับการวัดของระบบ LDS 26
การเชื่อมต่อวงจรรับส่งข้อมูลแบบขนาน (8255PPI) 27
กับวงจรขับกำลังสเตปปิ้งมอเตอร์
บทที่ 5 โครงสร้างของโปรแกรม 30
โปรแกรมควบคุมสำหรับฮาร์ดแวร์ 30
- ส่วนควบคุม XY-Table 30
- ส่วนการสแกนและเก็บข้อมูล 30
โปรแกรมการสร้างภาพและการแสดงผล 30
- การแสดงผลโดยการจำลองภาพ 3 มิติแบบธรรมดา 30
- การแสดงผลโดยการจำลองภาพแบบให้เฉดสีแสดงความสูงต่ำ 32
- การแสดงผลโดยการวิเคราะห์ภาคตัดขวางของพื้นผิววัตถุ 32
การใช้งานโปรแกรม Laser Surface Scanning System 33
- Scan 34
- Get Image 35
- Control Motor 36
- Information 36
- Quit 36
บทที่ 6 ผลการทดลอง 39
ขั้นตอนการทดลอง 39
ผลการทดลองและวิจารณ์ 40
- การหาค่า Resolution ของการวัดในแนวตั้ง 40
- การหาความละเอียดของการวัดในแนวนอน 41
ตัวอย่างภาพพื้นผิววัตถุที่ได้ทำการตรวจสอบ 43

บทที่ 7 สรุปและวิจารณ์ 45

ภาคผนวก ก โปรแกรม

ภาคผนวก ข Data Sheet

เอกสารอ้างอิง

ประวัติผู้ทำโครงงาน

สารบัญรูป

รูปที่ 1.1 แผนภาพแสดงการทำงานของ Laser Surface Scanning System 1

รูปที่ 2.1 แสดงลักษณะหัววัดพื้นผิวแบบเข็ม 4

รูปที่ 2.2 แสดงลักษณะการแสดงผลในการวัดพื้นผิวบน XY-Graphic recorder 5

รูปที่ 2.3 แสดงโครงสร้างภายในและหลักการทำงานของตัวตรวจวัดระยะชัดด้วยเลเซอร์ 6

รูปที่ 2.4 แสดงแผนภาพของสามเหลี่ยมทางแสง 6

รูปที่ 2.5 แสดงระบบเลนส์ที่ใช้ใน LDS 7

รูปที่ 2.6 แสดงปรากฏการณ์ของ PSD 8

รูปที่ 2.7 แผนภูมิของภาคประมวลสัญญาณ 9

รูปที่ 2.8 แสดงวงจรของภาคประมวลสัญญาณ 10

รูปที่ 2.9 กราฟแสดงความสัมพันธ์ระหว่างระยะทางกับศักดาไฟฟ้าขาออก 10

รูปที่ 2.10 กราฟแสดงคุณสมบัติการตอบสนองของเอาท์พุทต่อความเร็วในการวัด 11

รูปที่ 3.1 แสดง XY-Table ที่ใช้ในการสแกนวัตถุ 13

รูปที่ 3.2 บล็อกไดอะแกรมแสดงระบบสเตปปิ้งมอเตอร์ 14

รูปที่ 3.3 โครงสร้างของสเตปปิ้งมอเตอร์และลักษณะเส้นแรงแม่เหล็กที่เกิดขึ้น 16

รูปที่ 3.4 แสดงลำดับการส่งสัญญาณควบคุมมอเตอร์ 4 เฟสแบบกระตุ้นครั้งละเฟส 17

รูปที่ 3.5 แสดงลำดับการส่งสัญญาณควบคุมมอเตอร์ 4 เฟสแบบครึ่งสเตป 18

รูปที่ 3.6 ผลตอบสนองในการเข้าสู่สภาวะคงตัว 19

รูปที่ 4.1 สัญญาณอินเตอร์เฟสกับสัญญาณไมโครคอมพิวเตอร์ 20

รูปที่ 4.2 แผนผังวงจรภายในและการจัดขาของไอซี 8255 23

รูปที่ 4.3 การต่อ 8255 กับวงจร Decode 24

รูปที่ 4.4 บล็อกไดอะแกรมของ A/D Convertor แบบบันทึกการประมาณค่าอย่างต่อเนื่อง (SAR) 25

รูปที่ 4.5 แสดงวงจรเชื่อมต่อส่วนของวงจรแปลงสัญญาณอนาลอกเป็นดิจิตอล 27

รูปที่ 4.6 วงจรขับกำลังสเตปปิ้งมอเตอร์ 28

รูปที่ 5.1 แสดงโฟลว์ชาร์ทวิธีการเก็บข้อมูลที่ได้จากการสแกนในระนาบ XY 31

รูปที่ 5.2 แสดงภาพการแสดงพื้นผิวโดยการจำลองภาพ 3 มิติแบบธรรมดา 32

รูปที่ 5.3 แสดงภาพการแสดงพื้นผิวโดยการจำลองภาพแบบให้เฉดสีแสดงความสูงต่ำ 32
รูปที่ 5.4 แสดงภาพการแสดงพื้นผิววัตถุโดยการวิเคราะห์ภาคตัดขวาง 33
รูปที่ 5.5 แสดงภาพหน้าจอหลังจากเรียกโปรแกรมแล้ว 33
รูปที่ 5.6 แสดงภาพเมนูหลัก (Main Menu) 34
รูปที่ 5.7 แสดงภาพหน้าจอการสแกนเก็บข้อมูล 35
รูปที่ 5.8 แสดงภาพหน้าจอการนำข้อมูลมาทำการแสดงผล 36
รูปที่ 5.9 แสดงภาพหน้าจอการควบคุมมอเตอร์ 37
รูปที่ 5.10 แสดงภาพหน้าจอส่วนของ Information 37
รูปที่ 5.11 แสดงภาพหน้าจอการออกจากโปรแกรม 38
รูปที่ 6.1 แสดงการวางตำแหน่งอุปกรณ์ต่าง ๆ และแสดงเปรียบเทียบกับทิศตั้งแกน 39
รูปที่ 6.2 แสดงระยะการทำงานของ Laser Displacement Sensor 39
รูปที่ 6.3 แสดงการคำนวณหาค่า Resolution ของการวัดในแนวตั้ง 40
รูปที่ 6.4 แสดงภาคตัดขวางของส่วนของพื้นผิวทรงกลมที่นำมาคำนวณ 41
รูปที่ 6.5 แสดงภาพที่ได้จากการสแกนตาข่ายลวดขนาดเล็ก 42
รูปที่ 6.6 ด้านก้อยของเหรียญบาท 43
รูปที่ 6.7 ไมโครชิปของ Eprom 43
รูปที่ 6.8 เส้นแสดงสเกล 1/64 นิ้วของไม้บรรทัดเหล็ก 44
รูปที่ 6.9 พระเครื่อง (สมเด็จวัดระฆัง) 44

สารบัญตาราง

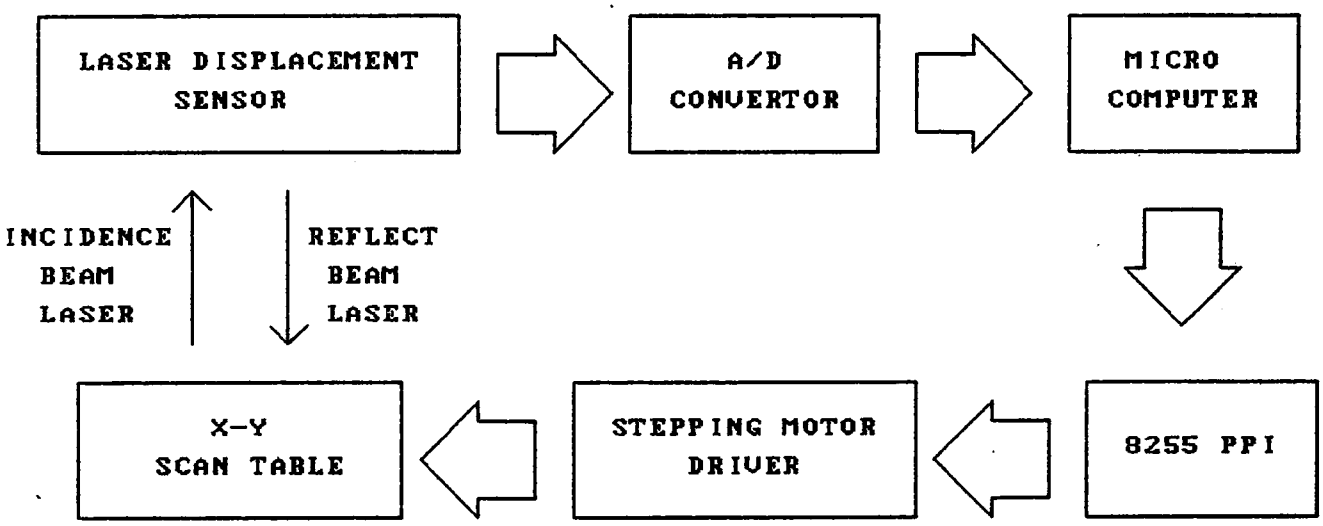
ตารางที่ 3.1 แสดงมุมสเตปสำหรับมอเตอร์ที่มีจำนวนสเตปต่อรอบค่าต่าง ๆ 15

ตารางที่ 4.1 การแบ่งแอดเดรสของ IBM/PC 21

บทที่ 1

บทนำ

ในปัจจุบันเทคโนโลยีทางด้านเลเซอร์ได้มีการพัฒนาไปอย่างมาก ได้มีการประยุกต์ใช้งานเลเซอร์อย่างกว้างขวาง ทั้งในด้านการแพทย์ งานทางวิทยาศาสตร์ อุตสาหกรรม และทางด้านความบันเทิง ในโครงงานพิเศษ Laser Surface Scanning System นี้ ได้ประยุกต์ใช้เลเซอร์ในการศึกษาพื้นผิวของวัตถุโดยทำการยิงแสงเลเซอร์ไปจนทั่วผิววัตถุ และทำการเก็บข้อมูลที่ได้เข้าคอมพิวเตอร์ทำการประมวลผลและสร้างภาพวัตถุใน 3 มิติขึ้น ซึ่งโครงงานพิเศษนี้มีส่วนประกอบที่สำคัญดังแสดงในรูปที่ 1.1



รูปที่ 1.1 แผนภาพแสดงการทำงานของ Laser Surface Scanning System

จากรูปจะเห็นว่า ในโครงงานพิเศษนี้ประกอบด้วยส่วนสำคัญ ๆ ดังนี้

1. Laser Displacement Sensor (LDS)
2. วงจรแปลงสัญญาณอนาลอกเป็นสัญญาณดิจิตอล
   (Analog To Digital Convertor)
3. การ์ด 8255 PPI
4. วงจรขับสเตปปิ้งมอเตอร์ (Stepping Motor Driver Circuit)
5. X-Y Scan Table

Laser Displacement Sensor (LDS)

เป็นแหล่งกำเนิดแสงเลเซอร์ และในขณะเดียวกันก็ทำงานเป็นตัว Displacement Sensor ภายในตัวโดยมีหลักการทำงานดังนี้ เมื่อยิงแสงเลเซอร์ไปยังวัตถุจะมีแสงสะท้อนกลับจากวัตถุสะท้อนกลับมายังตัว Displacement Sensor และให้สัญญาณออกมาเป็นสัญญาณอนาลอกซึ่งสัมพันธ์กับความตื้นลึกของผิววัตถุ ในโครงงานนี้ใช้ Laser Displacement Sensor ของ KEYENCE รุ่น LB-70

วงจรแปลงสัญญาณอนาลอกเป็นสัญญาณดิจิตอล (A/D Convertor)

เป็นวงจรซึ่งใช้แปลงสัญญาณอนาลอกที่ได้รับจาก Laser Displacement Sensor ไปเป็นสัญญาณดิจิตอล เพื่อนำไปเชื่อมต่อเข้ากับไมโครคอมพิวเตอร์ ในโครงงานพิเศษนี้ใช้ IC เบอร์ ADC574AJ ทำการแปลงสัญญาณอนาลอกเป็นสัญญาณดิจิตอลซึ่งมีการทำงานเป็นแบบการประมาณค่าทีละบิต (Successive Approximate ,SA)

การ์ด 8255 PPI

เป็นส่วนที่ใช้ในการเชื่อมต่อ วงจรขับสเตปปิ้งมอเตอร์กับไมโครคอมพิวเตอร์และใช้ในการควบคุมทิศทางการหมุนของสเตปปิ้งมอเตอร์ และเนื่องจากสเตปปิ้งมอเตอร์ทำงานโดยมีการกระตุ้นแบบเป็นเฟสจึงใช้พอร์ท A ของ 8255 ในการควบคุมทิศทางการหมุนของมอเตอร์

วงจรขับสเตปปิ้งมอเตอร์ (Stepping Motor Driver Circuit)

เป็นวงจรขับกำลังสเตปปิ้งมอเตอร์เพื่อทำให้สเตปปิ้งมอเตอร์สามารถหมุนแท่น X-Y Scan Table ได้

X-Y Scan Table

เป็นส่วนที่ใช้สำหรับวางวัตถุที่ต้องการจะสแกน การทำงานจะใช้สเตปปิ้งมอเตอร์ 2

ตัวในการหมุนแท่น โดยทำการหมุนในทิศทางในแนวแกน X และทิศทางในแนวแกน Y

วัตถุประสงค์ของโครงงานวิจัยนี้

1. เพื่อศึกษาการนำเอาแสงเลเซอร์มาประยุกต์ใช้งานในเชิงอุตสาหกรรม
2. เพื่อศึกษาการตรวจสอบพื้นผิวหน้าของวัตถุโดยใช้ Laser Displacement Sensor (LDS)
3. เพื่อศึกษาข้อจำกัดของการใช้ระบบสแกนพื้นผิวด้วยเลเซอร์
4. เพื่อศึกษาการนำข้อมูลมาวิเคราะห์และแสดงผลโดยใช้คอมพิวเตอร์กราฟิก

ขั้นตอนการดำเนินงาน

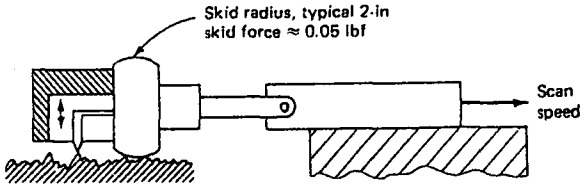
1. ศึกษาการทำงานของโครงงานในส่วนต่าง ๆ ดังนี้
   - Laser Displacement Sensor (LDS)
   - Stepping Motor
   - A/D Convertor
   - การ์ด 8255 และการเชื่อมต่อกับไมโครคอมพิวเตอร์
   - การเก็บข้อมูลและการควบคุม Stepping Motor
2. เขียนโปรแกรมซึ่งใช้ในการเก็บข้อมูล แสดงผลในลักษณะต่าง ๆ และควบคุมสเตปปิ้งมอเตอร์
3. ทดสอบการทำงานของโครงงาน ทำการสแกนเก็บข้อมูลเข้าไมโครคอมพิวเตอร์และนำข้อมูลที่เก็บได้มาทำการแสดงผล

บทที่ 2
การตรวจสอบพื้นผิววัตถุ

2.1 การศึกษาคุณสมบัติของพื้นผิววัตถุ

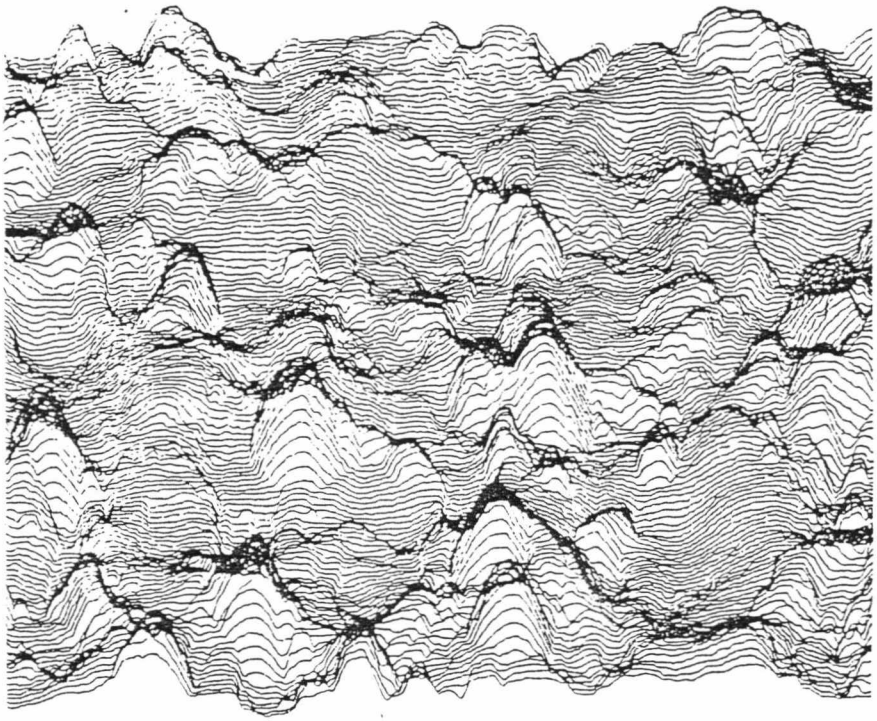
การตรวจสอบคุณสมบัติของพื้นผิววัตถุในระดับที่เล็กขนาดมองด้วยตาเปล่าไม่เห็นและไม่สามารถทำการวัดอย่างละเอียดด้วยวิธีธรรมดาได้มีความสำคัญอย่างมากในอุตสาหกรรมหลายอย่างเช่น การหารอยตำหนิบนพื้นผิวของสารกึ่งตัวนำ การขัดผิว การสึกหรอของวัสดุ การเคลือบสี การรั่วซึมในการปิดผนึก การสูญเสียเนื่องจากการกระเจิงของแสงบนพื้นผิววัตถุ ฯลฯ การตรวจสอบดังกล่าวจะนำไปสู่การปรับปรุงการผลิตได้

การตรวจสอบพื้นผิวโดยมากจะใช้เครื่องมือที่เป็นเข็มวัดชนิดเป็นเพชรโดยมีรัศมีประมาณ 0.0001 นิ้วโดยใช้แรงกด 0.0001 ถึง 0.001 ปอนด์/ฟุต สำหรับในระบบที่มีความละเอียดมาก ๆเข็มวัดจะมีรัศมีเพียง 0.00005 นิ้วและลดแรงกดลง ขนาดของเข็มวัดเป็นปัจจัยสำคัญที่มีผลต่อความละเอียดของการวัดในแนวนอน ในการวัดความขรุขระของแนวพื้นผิวที่เลือกไว้ เข็มวัดจะกระดกขึ้นลงตามความขรุขระโดยเคลื่อนที่ไปด้วยความเร็วที่กำหนดไว้(0.0001 นิ้วถึง 0.1 นิ้วต่อวินาที) เข็มวัดจะติดตั้งบนแคร่เลื่อนที่มีตัวแปลงสัญญาณจากการกระดกขึ้นลงของเข็มวัด



รูปที่ 2.1 แสดงลักษณะหัววัดพื้นผิวแบบเข็ม

การแสดงผลของการวัดแบบนี้จะใช้ XY-Graphic Recorder เป็นตัวรับสัญญาณจากตัวแปลงสัญญาณ (Displacement transducer) นำมาแสดงให้เห็นในรูปอัตราการขยายระหว่างการเคลื่อนที่ของเข็มวัดต่อการเคลื่อนที่ของปากกา Recorder สำหรับแกนตั้ง (ความขรุขระ) จะขยายขึ้นประมาณ 1 ล้านเท่า ขณะที่ความละเอียดในแนวนอนจะขยายขึ้นประมาณ 500 เท่า การแปลงโดยตรงเช่นนี้ทำให้สามารถเข้าใจในลักษณะของพื้นผิวได้



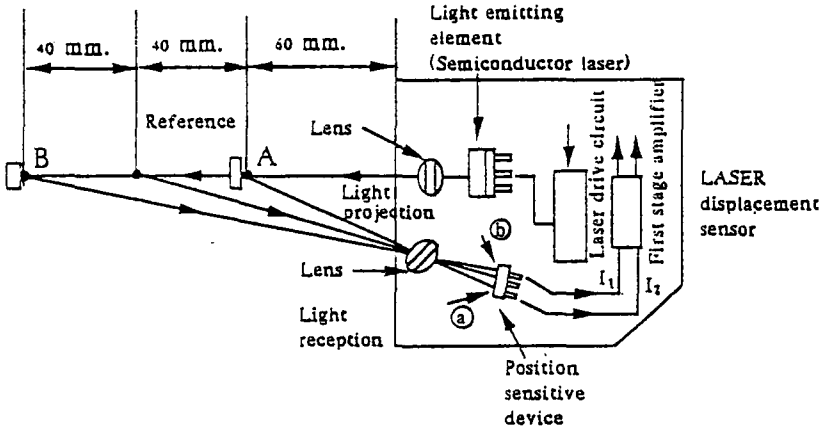
รูปที่ 2.2 แสดงลักษณะการแสดงผลในการวัดพื้นผิวบน XY-Graphic Recorder

อย่างไรก็ตาม เครื่องมือที่ใช้หลักการทางแสงดูจะเป็นประโยชน์อย่างยิ่ง เมื่อเราต้องการไม่ให้มีการสัมผัสกับพื้นผิวหน้าของวัตถุตัวอย่างโดยตรง ซึ่งจะทำความเสียหายให้กับพื้นผิววัตถุได้

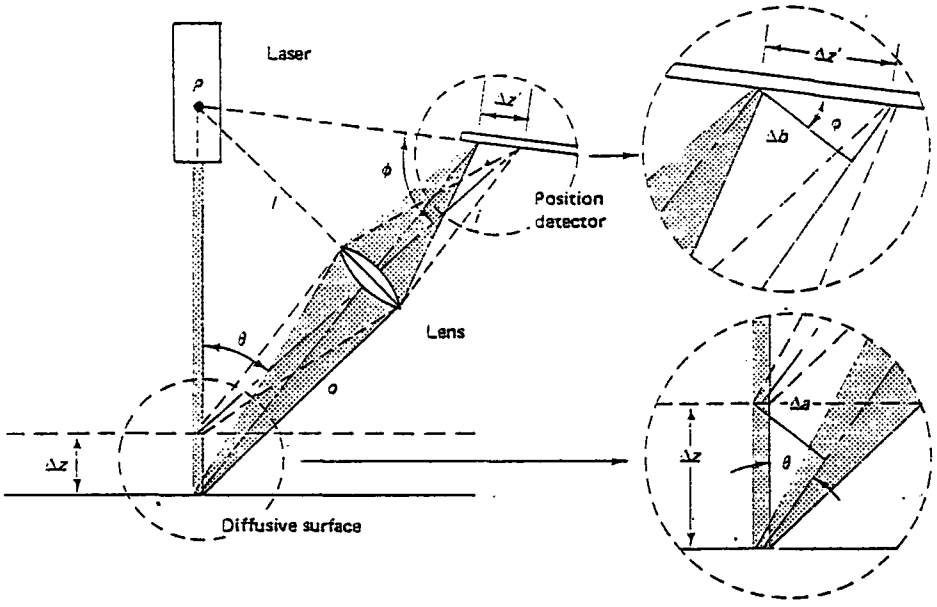
2.2 ตัวตรวจวัดระยะขจัดด้วยเลเซอร์ (Laser Displacement Sensor)

ตัวตรวจวัดระยะขจัดด้วยเลเซอร์(Laser Displacement Sensor)ชนิดนี้ใช้หลักการสามเหลี่ยมทางแสง (Optical Triangulation) ซึ่งเป็นวิธีการหาระยะตื้นลึกของพื้นผิววัตถุโดยไม่ต้องสัมผัสกับวัตถุมีหลักการดังนี้ ปล่อยลำแสงเลเซอร์ชนิดสารกึ่งตัวนำออกมาผ่าน Projector Lens ไปตกกระทบวัตถุโดยทิศทางของแสงเลเซอร์ตกกระทบตั้งฉากกับพื้นผิววัตถุ ดังนั้นลำแสงเลเซอร์ส่วนหนึ่งที่เกิดการกระเจิง (Scattering) กลับไปตกกระทบบน Position Sensitive Device (PSD) ซึ่งเป็นโฟโตไดโอดชนิดพิเศษแบบ PIN Diode ลำแสงส่วนที่กระเจิงกลับจะตกกระทบบน PSD ณ ที่ตำแหน่งใด ๆ ซึ่งสัมพันธ์กับระยะทางความลึกของพื้นผิวของวัตถุ พิจารณาจากรูปที่ 2.3 เมื่อลำแสงเลเซอร์ตกกระทบพื้นผิววัตถุที่ตำแหน่ง A จะเห็นว่าลำแสงมีการกระเจิงออกจะมีส่วนหนึ่งตกกระทบ PSD ณ ที่ตำแหน่ง (a) ส่วนวัตถุที่ตำแหน่ง B จะให้ลำแสงตกกระทบที่ PSD ที่ตำแหน่ง (b) ซึ่งตำแหน่งที่ลำแสงตกกระทบที่ PSD ทั้ง (a) และ (b) จะให้ความสัมพันธ์กับระยะทางความลึกของพื้นผิว ดังนั้นจะเห็นว่าตำแหน่งที่ลำแสงตกกระทบ PSD จะเปลี่ยนแปลงความ

สัมพันธ์ไปกับระยะทางความลึกพื้นผิว



รูปที่ 2.3 แสดงโครงสร้างภายในและหลักการทำงานของตัวตรวจวัดระยะขจัดด้วยเลเซอร์



รูปที่ 2.4 แสดงแผนภาพของสามเหลี่ยมทางแสง

พิจารณารูปที่ 2.4 เป็นภาพอธิบายการหาความสัมพันธ์ของระยะความลึกพื้นผิวโดยวิธีสามเหลี่ยมทางแสงแบบจุดแสงซึ่งเป็นการจัดวางระนาบของพื้นผิววัตถุ ระนาบของเลนส์

และของตัวตรวจจับตำแหน่งให้สัมพันธ์กันโดยใช้จุด P เป็นจุดอ้างอิง

เมื่อ θ คือ มุมระหว่างลำแสงเลเซอร์กับ Optical axis ของเลนส์

ϕ คือ มุมระหว่าง Optical axis กับระนาบของตัวตรวจจับตำแหน่ง

จะได้ความสัมพันธ์ระหว่างระยะวัตถุ o กับระยะสมมติ i คือ

$$l = i \tan \phi \qquad (2.1)$$

โดย l คือ ระยะระหว่างจุด P กับจุดศูนย์กลางของเลนส์

ดังนั้นมุม ϕ จะหาได้จาก

$$\tan \phi = (1/|M|) \tan \theta \quad ;m = -i/o \qquad (2.2)$$

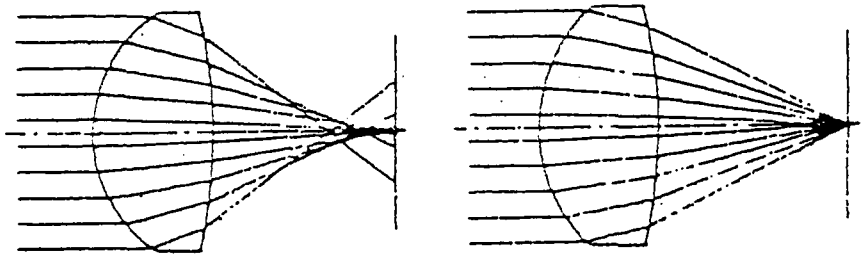
เมื่อ M คือ กำลังขยายของเลนส์

สมมติให้ระยะพื้นผิวอ้างอิงกับระยะพื้นผิวของวัตถุเท่ากับ ΔZ พิจารณารูปขยายทางด้านขวาของรูป 2.4 ซึ่งแสดงลำแสงที่ตกกระทบลงบนวัตถุจะได้ $\Delta a = \Delta Z \sin\theta$ และลำแสงที่ตกกระทบลงบนตัวตรวจจับตำแหน่งได้ $\Delta b = \Delta Z^* \sin\phi$ เมื่อเปลี่ยนขนาดกำลังขยายของเลนส์ $|m|$ ให้อยู่ในรูป $|m| = \Delta b/\Delta a$ จากสมการเหล่านี้สามารถหาระยะทางของจุดแสงเลเซอร์ 2 จุดบนตัวตรวจจับตำแหน่งได้

$$\Delta Z^* = |m| (\sin\theta/\sin\phi) \Delta Z \qquad (2.3)$$

2.3 ระบบเลนส์ที่ใช้

ระบบเลนส์ที่ใช้ใน LDS ต้องเป็นระบบเลนส์ที่มีความเที่ยงตรงสูงมาก ในที่นี้เป็นระบบ Aspherical Lens เพราะระบบเลนส์ชนิดนี้มีความเที่ยงตรงและให้ความละเอียดที่สูงกว่าระบบ Sperical Lens แสดงดังรูปที่ 2.5



(a) Spherical lens (b) Aspherical lens

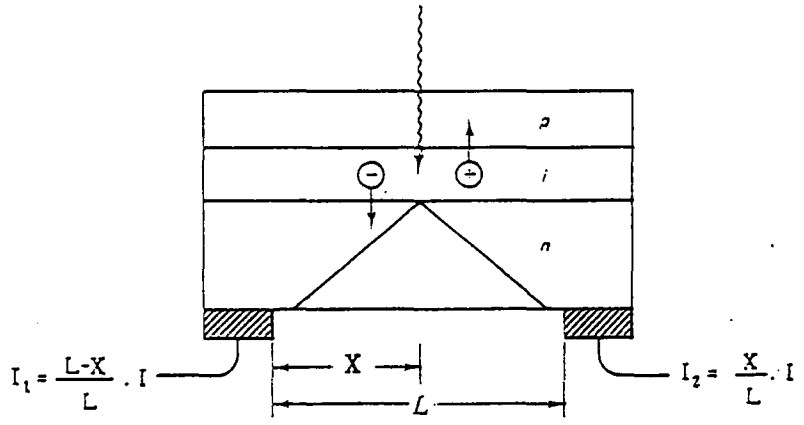
รูปที่ 2.5 แสดงระบบเลนส์ที่ใช้ใน LDS

2.4 ตัวตรวจจับตำแหน่ง

2.4.1 ตัวตรวจจับที่ไวต่อตำแหน่ง (Position Sensitive Device ;PSD)

เป็นที่ทราบกันแล้วว่าเมื่อรอยต่อพี-เอ็นของสารกึ่งตัวนำ เมื่อได้รับการฉายแสงที่มีความยาวคลื่นที่เหมาะสมจะกระตุ้นให้เกิดปรากฏการณ์โฟโตอิเลคตริค (Photoelectric effect) ขณะเดียวกันก็จะเกิดศักดาแสง(Photovoltage) ตกคร่อมด้านทั้งสองของรอยต่อเมื่อต่อขั้วไฟฟ้าของรอยต่อ เมื่อต่อขั้วไฟฟ้าของด้านทั้งสองเข้าหากันจะเกิดกระแสไหลได้ ในปี ค.ศ.1957 Wallmark ได้ค้นพบปรากฏการณ์โฟโตในแนวด้านข้าง (Lateral photo effect) นอกเหนือจากปรากฏการณ์โฟโตในแนวขวาง (Transverse photo effect) ซึ่งเป็นที่รู้จักกันดีโดยมีเงื่อนไขว่าแสงที่ตกกระทบจะมีการส่องสว่างไม่สม่ำเสมอทั่วพื้นที่รับแสง (Nonuniform illumination) ซึ่งจะตรวจจับได้จากศักดาแสงด้วยด้านข้างที่ปรากฏบนขั้วไฟฟ้าบนด้านเดียวกันกับพื้นผิวรับแสงขนาดใหญ่ของรอยต่อพี-เอ็น ถ้าวัสดุที่ใช้ทำมีความสมมาตรและแสงตกกระทบเต็มพื้นที่และมีความสม่ำเสมอ ศักดาแสงด้านข้างที่เกิดขึ้นจะสมดุลพอดี ดังนั้นจึงต้องบังคับให้แสงตกกระทบเป็นจุดเล็ก ๆ ซึ่งปรากฏการณ์ดังกล่าวสามารถสร้างอุปกรณ์นี้เป็นตัวตรวจจับตำแหน่งที่แม่นยำได้

อุปกรณ์ PSD เป็นรอยต่อพี-เอ็นที่มีพื้นที่ขนาดใหญ่โดยสร้างชั้นพีให้มีสภาพต้านทานสูงบนฐานรองชนิดเอ็นซึ่งมีสภาพต้านทานต่ำ การเพิ่มชั้น Intrinsic(i) ซึ่งมีสภาพต้านทานสูงลงไปจะทำให้ตัว PSD มีความไวต่อแสงที่ตกกระทบเพิ่มขึ้น เราเรียกโครงสร้างแบบนี้ว่า PIN ที่บนชั้นเอ็นจะมีขั้วไฟฟ้าแบบโอห์มมิกไว้



รูปที่ 2.6 แสดงปรากฏการณ์ของ PSD

เมื่อแสงเป็นจุดซึ่งใช้ในการชี้ตำแหน่งตกกระทบบนชั้นพีจะกระตุ้นให้เกิดคู่อิเลคตรอน-โฮลอิสระขึ้น เมื่อมันเดินทางไปถึงบริเวณรอยต่อก็จะถูกสนามไฟฟ้าผลักออกจากกันโดยจะเหลือโฮลส่วนเกินในชั้นพีและอิเลคตรอนส่วนเกินในชั้นเอ็น ทั้งอิเลคตรอนและโฮลอิสระเหล่านี้จะเคลื่อนที่ออกไปในทุกทิศทุกทางห่างออกจากจุดที่แสงตกกระทบทำให้เกิดศักดาด้านข้างซึ่งขึ้นกับระยะทางที่ประจุอิสระเคลื่อนที่ออกไป ให้ความสัมพันธ์ของกระแสที่ได้จากการตรวจจับสัญญาณจากตำแหน่งที่แสงสะท้อนจากวัตถุตกกระทบลงบน PSD ซึ่งวัดกระแสเทียบกับตำแหน่งได้ดังนี้

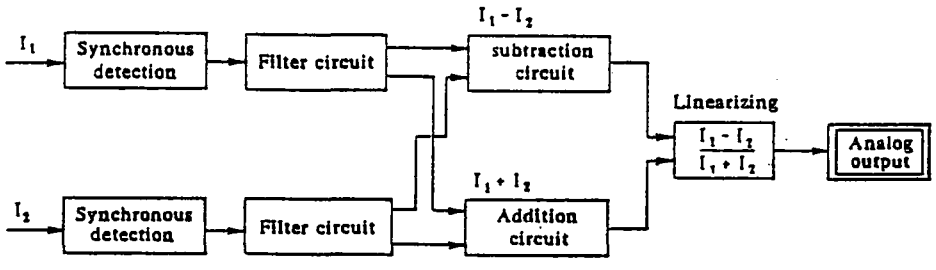
$$I_1 = (L-X)I/L \qquad (2.4)$$

$$I_2 = (X/L)I \qquad (2.5)$$

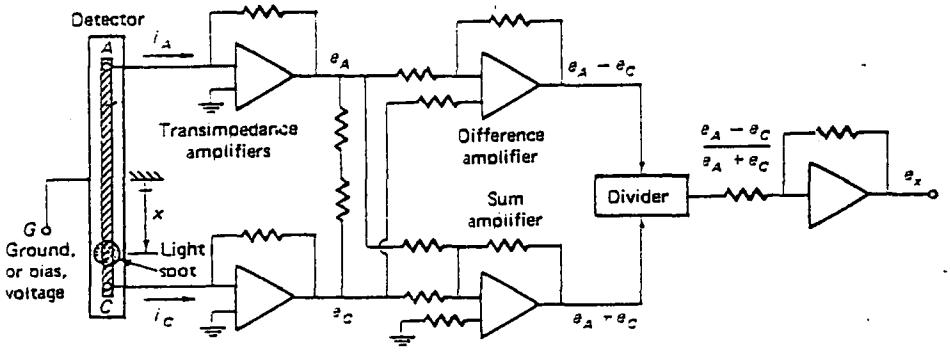
$$(I_1-I_2)/(I_1+I_2) = (L-2X)/L \qquad (2.6)$$

2.4.2 ระบบการประมวลสัญญาณ

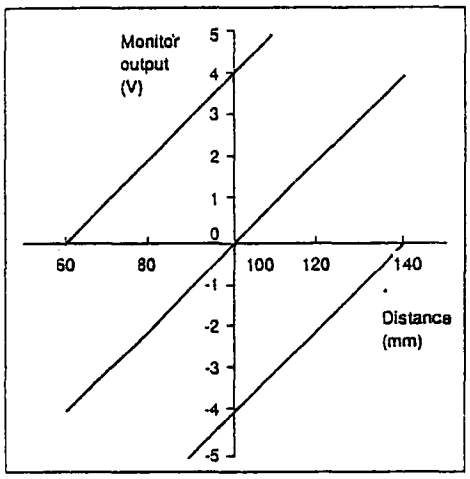
จากรูป 2.6 เมื่อแสงตกกระทบบน PSD จะให้กระแสเอาท์พุท 2 ค่าคือ I_1 และ I_2 ซึ่งกระแสเอาท์พุท I_1, I_2 เป็นฟังก์ชั่นเชิงเส้นกับค่าของ X รูป 2.7 เป็นการแสดงแผนภูมิของภาคประมวลสัญญาณ พิจารณาจากรูป 2.8 การที่ (I_1-I_2) ถูกหารด้วย (I_1+I_2) เพื่อทำให้ค่ากระแสเป็นเชิงเส้นกับระยะทางโดยไม่ขึ้นกับค่าความเข้มแสงที่กระเจิงกลับ



รูปที่ 2.7 แสดงแผนภูมิของภาคประมวลสัญญาณ



รูปที่ 2.8 แสดงวงจรของภาคประมวลสัญญาณ



รูปที่ 2.9 กราฟแสดงความสัมพันธ์ระหว่างระยะทางกับศักดาไฟฟ้าขาออก

2.5 คุณสมบัติของเลเซอร์ดิสเพลซเมนต์เซ็นเซอร์

ในโครงงานวิจัยนี้ได้เลือกใช้เลเซอร์ดิสเพลซเมนต์เซ็นเซอร์ของบริษัท Keyence รุ่นLB-70 ซึ่งเป็นแหล่งกำเนิดแสงเลเซอร์แบบเซมิคอนดัคเตอร์และตัวตรวจจับระยะความต้นลึกของพื้นผิววัตถุในเครื่องเดียวกัน ซึ่งแสดงข้อมูลคุณสมบัติของเลเซอร์ดิสเพลซเมนต์เซ็นเซอร์ดังนี้

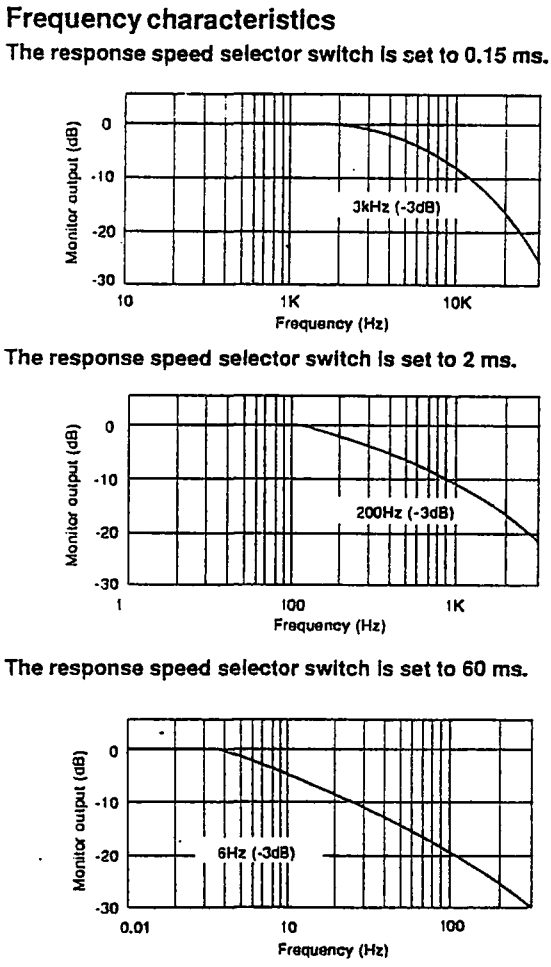
2.5.1 ช่วงระยะการวัด

LB-70 มีระยะตรวจจับความต้นลึกของพื้นผิววัตถุ +40 mm.ถึง -40 mm. ที่ให้ค่าสัญญาณอนาลอกโวลเตจ +4 V ถึง -4 V (0.1 V/mm.) ซึ่งจะมีความสัมพันธ์ดังกราฟใน

รูปที่ 2.9

2.5.2 สัญญาณเอาท์พุทกับความเร็วของการวัด

LB-70 จะมีสวิทช์ให้เลือกว่าใช้ความเร็วในการวัดเท่าใด ถ้าเลือกใช้ไม่เหมาะกับความเร็วของการเคลื่อนที่ของวัตถุ จะทำให้ค่าเอาท์พุทไม่เป็นสัดส่วนโดยตรงกับระยะทาง แสดงคุณสมบัติการตอบสนองของการวัดของเอาท์พุทต่อความเร็วในการวัดเป็นกราฟดังรูปที่ 2.10



รูปที่ 2.10 กราฟแสดงคุณสมบัติการตอบสนองของเอาท์พุทต่อความเร็วในการวัด

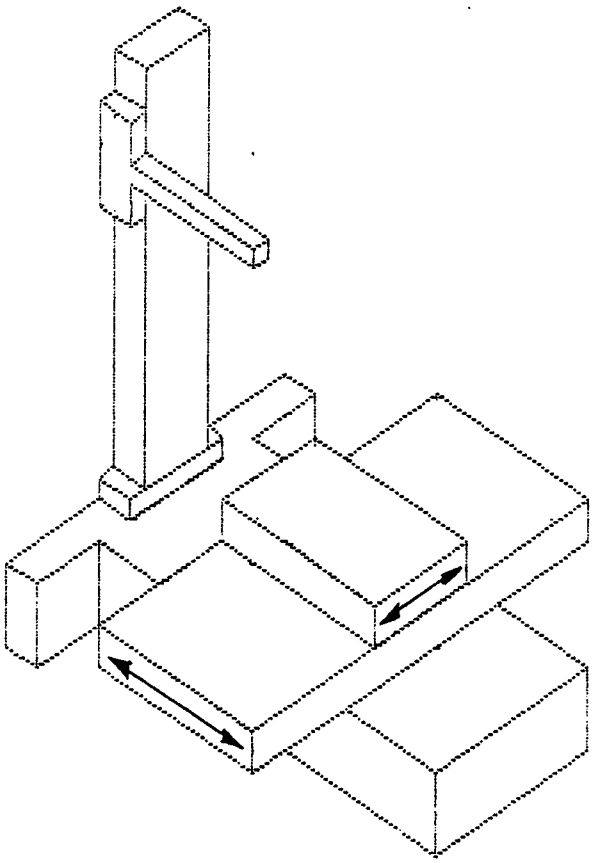
2.5.3 ค่าความละเอียดในการวัด

ตามคู่มือการใช้ LB-70 ระบุว่าสามารถที่จะวัดพื้นผิวได้ความละเอียด 10 μm ที่ความเร็ว 500 ms 40 μm ที่ 20 ms และ 180 μm ที่ความเร็ว 0.7 ms แสดงให้เห็นว่าเมื่อต้องการให้การวัดมีความละเอียดและถูกต้องมาก ควรจะมีสวิทช์เลือกใช้ความเร็ว

500 ms ซึ่งก็คือความเร็วที่ช้าลงในการแปลงค่าจากระยะทางให้เป็นค่าเอาท์พุทโวลเตจ ดังนั้นจะได้ค่าความละเอียดเพิ่มขึ้น แต่เวลาที่ใช้ก็มากขึ้นตามไปด้วย

บทที่ 3
ระบบขับเคลื่อนวัตถุ

เป็นอุปกรณ์ที่ทำหน้าที่ขับเคลื่อนวัตถุที่จะทำการวัดความตื้นลึกของพื้นผิว โดยใช้สเตปปิ้งมอเตอร์ 2 ตัวเป็นตัวขับเคลื่อนให้แท่นเคลื่อนที่ทั้งในแนวแกน X และแกน Y แสดงได้ดังรูปที่ 3.1



รูปที่ 3.1 แสดง XY-Table ซึ่งใช้ในการสแกนวัตถุ

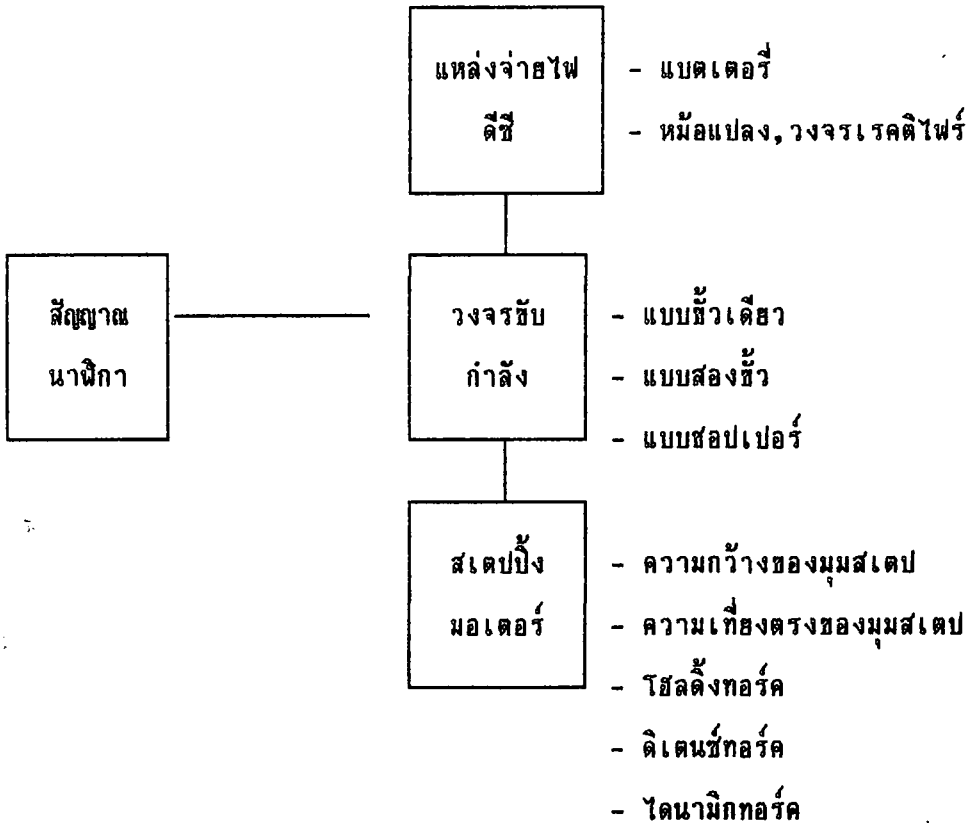
3.1 สเตปปิ้งมอเตอร์

สเตปปิ้งมอเตอร์เป็นมอเตอร์ไฟฟ้าชนิดหนึ่งซึ่งออกแบบมาเป็นพิเศษโดยสามารถควบคุมความเร็วได้ด้วยการควบคุมดีซีพัลส์ที่ใช้ในการขับเคลื่อนสเตปปิ้งมอเตอร์ ซึ่งลักษณะการป้อนกระแสไฟฟ้าเป็นแบบสวิทชิ่ง

กล่าวได้ว่าสเตปมอเตอร์เป็นตัวทำหน้าที่เปลี่ยนสัญญาณข้อมูลแบบดิจิตอล (Digital information) ไปสู่การเคลื่อนที่ทางกลอย่างได้สัดส่วนกัน โดยที่แกนหมุนหรือโรเตอร์ของมอเตอร์ชนิดนี้มักถูกควบคุมให้หมุนเป็นสเตปอย่างเที่ยงตรง

ระบบสเตปปิ้งมอเตอร์

การเลือกใช้สเตปปิ้งมอเตอร์เพื่อนำไปใช้ประโยชน์ในงานบางอย่างนั้น จะต้องมีความเข้าใจคุณลักษณะของมอเตอร์และวงจรที่ใช้ขับมอเตอร์ในรูปที่ 3.2 แสดงบล็อกไดอะแกรมของระบบสเตปปิ้งมอเตอร์



รูปที่ 3.2 บล็อกไดอะแกรมแสดงระบบสเตปปิ้งมอเตอร์

ความถูกต้องเที่ยงตรงของมุมสเตป ขณะที่ไม่มีโหลดจะถูกระบุสำหรับมอเตอร์แต่ละชนิดเช่น มอเตอร์ที่มีมุมสเตป 7.5 องศา ความผิดพลาดบวกลบ 10 ลิปดา ขณะเคลื่อนไป 1 สเตป เป็นต้น มอเตอร์ที่มีจำนวนสเตปต่อรอบเท่ากับ 4 จะมีค่าผิดพลาดน้อยมากเมื่อหมุนครบ 1 รอบ เพราะขณะที่หมุนมา ณ ตำแหน่งเดิมขณะเริ่มต้น ขั้วแม่เหล็กและฟลักซ์แม่เหล็ก (Flux) คงเดิม ด้วยเหตุนี้การเปลี่ยนตำแหน่งของสเตปปิ้งมอเตอร์ที่ต้องการความถูกต้องสูง ๆ จะต้องแบ่งจำนวนสเตปต่อรอบเป็นจำนวนเท่าของ 4 สเตปเพื่อลดการสะสมของค่าผิดพลาด (Step angle error)ซึ่งเป็นรูปแบบการทำงานแบบ 4 สเตป ตัวอย่างของมุมสเตปแสดงดังตารางที่ 3.1

ตารางที่ 3.1 แสดงมุมสเตปสำหรับมอเตอร์ที่มีจำนวนสเตปต่อรอบค่าต่าง ๆ

มุมสเตป (องศา)	จำนวนสเตปต่อรอบ
0.90	400
1.80	200
3.60	100
3.75	96
7.50	48
15.00	24

แรงบิด (Torque)

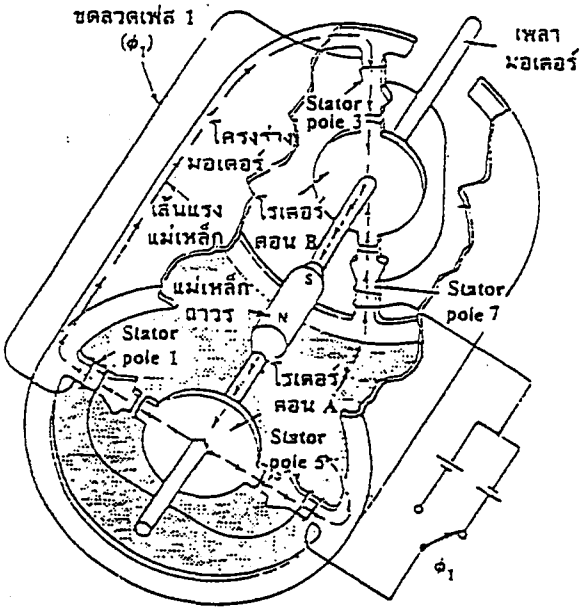
การทำงานของสเตปปิ้งมอเตอร์มีแรงบิดเกี่ยวข้องอยู่ 3 ชนิด คือ

โฮลดิ้งทอร์ค (Holding torque) คือแรงบิดที่ทำให้สเตปปิ้งมอเตอร์เริ่มหมุนไป 2 สเตปจากขณะหยุดนิ่ง ถ้าแรงบิดที่ให้สเตปปิ้งมอเตอร์มีขนาดมากกว่าระดับโฮลดิ้งทอร์ค จะทำให้มอเตอร์สูญเสียการหมุนแบบสเตปกลายเป็นการหมุนแบบต่อเนื่อง โดยปกติแรงบิดขณะทำงานของมอเตอร์จะน้อยกว่าระดับโฮลดิ้ง

ดิเทนท์ทอร์ค (Detent torque) สเตปปิ้งมอเตอร์แบบไฮบริด และแบบแม่เหล็กถาวรจะมีส่วนประกอบของโรเตอร์เป็นแม่เหล็กถาวร ซึ่งจะสร้างแรงบิดมาเบรคการหมุน

ของมอเตอร์อย่างสม่ำเสมอในขณะที่ไม่มีการป้อนกระแสเข้าขดสเตเตอร์ แรงบิดดังกล่าวนี้เรียกว่า ดิเตนช์ทอร์ค

ไดนามิกทอร์ค (Dynamic or Working torque) คือแรงบิดขณะทำงานซึ่งอาจเกิดการเปลี่ยนแปลงได้ เนื่องมาจากการปรับเปลี่ยนอัตราเร็วของมอเตอร์ โดยปกติการเปลี่ยนอัตราเร็วของมอเตอร์จะอยู่ในย่านระหว่างเส้นโค้งพูลอิน (Pull-in curve) และเส้นโค้งพูลเอาท์ (Pull-out curve) เพราะถ้าปรับอัตราเร็ว ณ จุดนอกโค้งพูลเอาท์ มอเตอร์จะสูญเสียการหมุนแบบเป็นสเตปได้หรือเกิดการหมุนแบบต่อเนื่องนั่นเอง



รูปที่ 3.3 โครงสร้างของสเตปปิ้งมอเตอร์และลักษณะเส้นแรงแม่เหล็กที่เกิดขึ้น

ฉะนั้นในโครงงานพิเศษนี้จึงได้เลือกใช้สเตปปิ้งมอเตอร์เป็นตัวขับเคลื่อนซึ่งสเตปปิ้งมอเตอร์ที่สามารถหาซื้อได้โดยทั่วไปเป็นแบบ 200 สเตป นั่นคือมอเตอร์ต้องเคลื่อนที่ไป 200 สเตปจึงหมุนได้ครบ 1 รอบหรือ 360 องศา ดังนั้นในแต่ละสเตปที่มอเตอร์หมุนแกนไปจะมีการเปลี่ยนแปลงมุมเท่ากับ 1.8 องศา (360 องศา / 200 สเตป)

3.2 ระบบขับเคลื่อน

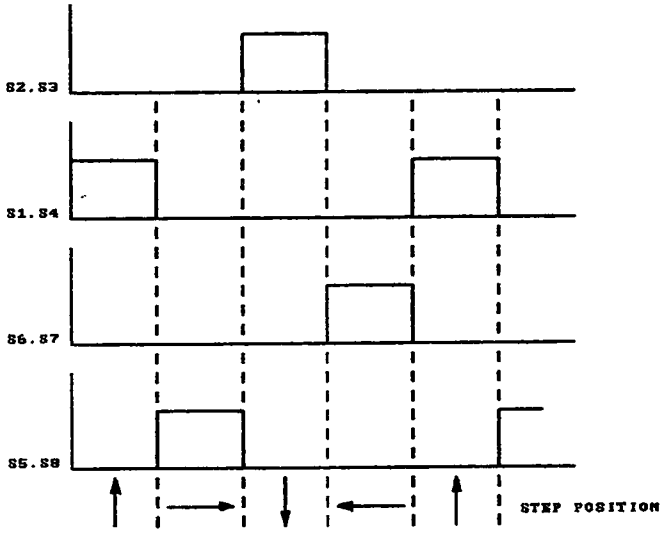
เนื่องจากโครงงานนี้มีวัตถุประสงค์เพื่อทำการสแกนพื้นผิวของวัตถุและต้องการความเที่ยงตรงแม่นยำของตำแหน่งสูง ดังนั้นจึงเลือกใช้ระบบขับเคลื่อนแบบ Lead Screw ที่มีระยะ Pitch เท่ากับ 0.5 mm.

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

3.3 ระบบควบคุม

การออกแบบระบบควบคุมแท่นสแกน XY ที่เหมาะสม ซึ่งก่อนอื่นต้องเข้าใจถึงการทำงานและคุณสมบัติบางประการของสเตปปิ้งมอเตอร์ก่อน สเตปปิ้งมอเตอร์โดยทั่วไปจะมีข้อมูลติดมากับตัวเสมอ ข้อมูลที่สำคัญได้แก่แรงดันต่อเฟสและค่าของมุมที่หมุนไปต่อสเตป ในที่นี้มอเตอร์ที่ใช้มีค่า 1.8 องศาต่อสเตป ดังนั้นถ้าระยะระหว่างเกลียวของบอลสกรูมีขนาด 2 mm. จะได้ว่ามีการเคลื่อนที่ 0.01 mm.หรือ 10 μm ต่อสเตป

3.3.1 หลักการขับมอเตอร์แบบกระตุ้นเฟสเดียว



รูปที่ 3.4 แสดงลำดับการส่งสัญญาณควบคุมมอเตอร์ 4 เฟสแบบกระตุ้นครั้งละเฟส

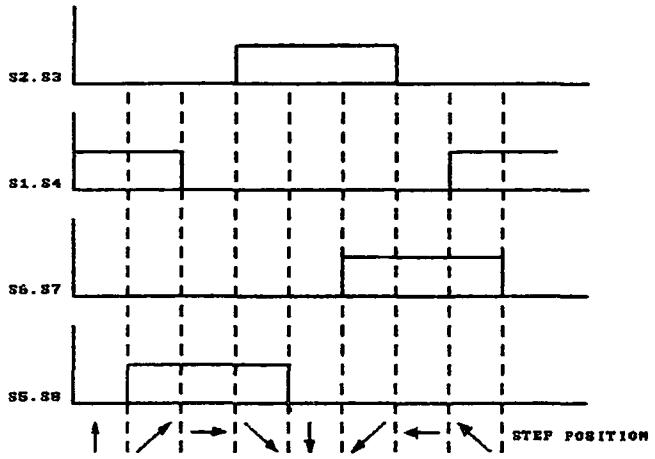
ในการขับสเตปปิ้งมอเตอร์ทั่วไปนั้น การขับ 1 สเตปก็คือการเลื่อนการจ่ายกระแสจากเฟสหนึ่งไปยังอีกเฟสหนึ่ง ซึ่งในมอเตอร์ส่วนใหญ่จะมีอยู่ 4 เฟสดังนั้นเมื่อเลื่อนถึงเฟสสุดท้ายก็จะเริ่มเฟสแรกใหม่ ซึ่งจะทำให้มอเตอร์หมุนต่อไปได้เรื่อย ๆ ดังรูปที่ 3.4

การขับสเตปปิ้งมอเตอร์แบบนี้จะทำให้ค่าแรงบิดลดลง 30 % แต่สามารถชดเชยค่าแรงบิดให้เพิ่มขึ้นได้โดยเพิ่มแรงดันของแหล่งจ่ายไฟซึ่งจะทำให้ประสิทธิภาพสูง แต่มีความแม่นยำของสเตปน้อย

3.3.2 หลักการขับมอเตอร์แบบกระตุ้นหนึ่งและสองเฟส

เป็นวิธีการขับแบบครึ่งสเตป ด้วยวิธีการนี้ทำให้สามารถเพิ่มความละเอียดของการ

เคลื่อนที่ให้มีความละเอียดเพิ่มขึ้นเป็นเท่าตัวโดยมอเตอร์จะหมุนไปครั้งละครึ่งสเตปเท่ากับ 0.9 องศา ผลที่ได้รับขณะใช้งานคือ แรงบิดจะมากและน้อยสลับกันไปเพราะถูกขับ 2 เฟส และ 1 เฟสสลับกัน ดังนั้นจะทำให้ไม่เกิดการออสซิลเลต แสดงดังรูปที่ 3.5

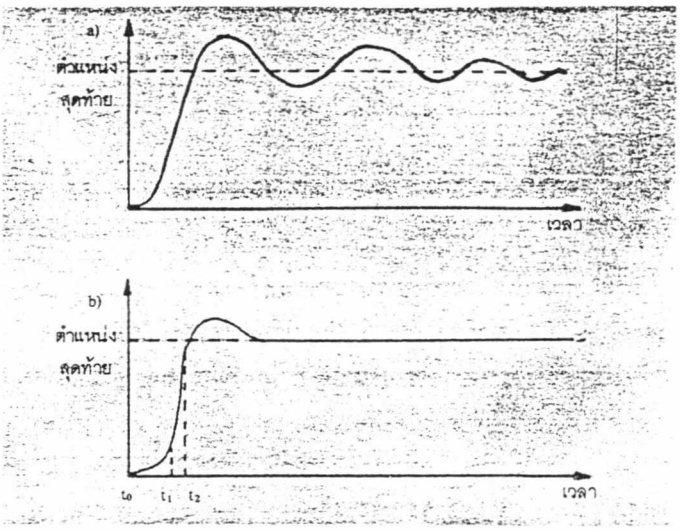


รูปที่ 3.5 แสดงลำดับการส่งสัญญาณควบคุมสเตปปิ้งมอเตอร์ 4 เฟสแบบครึ่งสเตป

3.3.3 การแกว่งเข้าสู่สภาวะคงตัว (Over shoot)

ขณะที่มอเตอร์หมุนไปในแต่ละสเตปและหยุด ณ สเตปใด ๆ จะเกิดการแกว่งหรือสั่นของโรเตอร์เข้าสู่ตำแหน่งสุดท้ายนั้น ๆ และใช้เวลาช่วงเวลาหนึ่งในการเข้าสู่สภาวะคงตัว แสดงเปรียบเทียบได้ดังรูป 3.6 (a) ซึ่งเป็นพฤติกรรมปกติของระบบที่ใช้สัญญาณพัลส์ ทั้งนี้ขึ้นอยู่กับโหลดและกำลังงานที่ได้รับจากภาคขับมอเตอร์เป็นคุณสมบัติที่สำคัญสมควรคำนึงถึงเพราะในการวัดพื้นผิวแต่ละจุดนั้นต้องให้มีความรวดเร็วในการหยุดนิ่งเพื่อความรวดเร็วและแม่นยำในการวัด อย่างไรก็ตามผลตอบสนองที่เกิดขึ้นนี้สามารถเปลี่ยนแปลงได้ คือลดเวลาในการเข้าสู่สภาวะคงตัวได้โดยการเพิ่มโหลดที่เป็นแรงเสียดทานเข้าไปในระบบ ซึ่งเป็นลักษณะการแก้ไขทางกลเช่น การใช้เครื่องต่อกำลังไปเพลาโดยใช้ความฝืดไม่ใช้เฟือง

วิธีการแก้ไขทางไฟฟ้า ทำได้โดยหน่วงสัญญาณพัลส์ลูกสุดท้ายในขบวนพัลส์ทั้งหมดโดยอาจถูกเปลี่ยนเป็น 3 ส่วนด้วยกันดังรูปที่ 3.6 (b) โดยส่วนแรกที่เวลา t_0 จะเป็นฟอร์เวิร์ดพัลส์ เวลา t_1 จะป้อนเป็นรีเวิร์สพัลส์เพื่อชลอการหมุนของโรเตอร์ และส่วนสุดท้ายที่เวลา t_2 จะเป็นฟอร์เวอร์สพัลส์ป้อนเพื่อให้โรเตอร์หยุด ณ ตำแหน่งที่ต้องการซึ่งวิธีการนี้จะสร้างแรงบิดน้อยกว่าปกติ ดังนั้นจึงลดการสั่นหรือแกว่งของเพลาได้



รูปที่ 3.6 ผลตอบสนองในการเข้าสู่สภาวะคงตัว

ก) ผลตอบสนองเมื่อมีการหน่วงน้อย

ข) ผลตอบสนองเมื่อมีการหน่วงทางไฟฟ้า

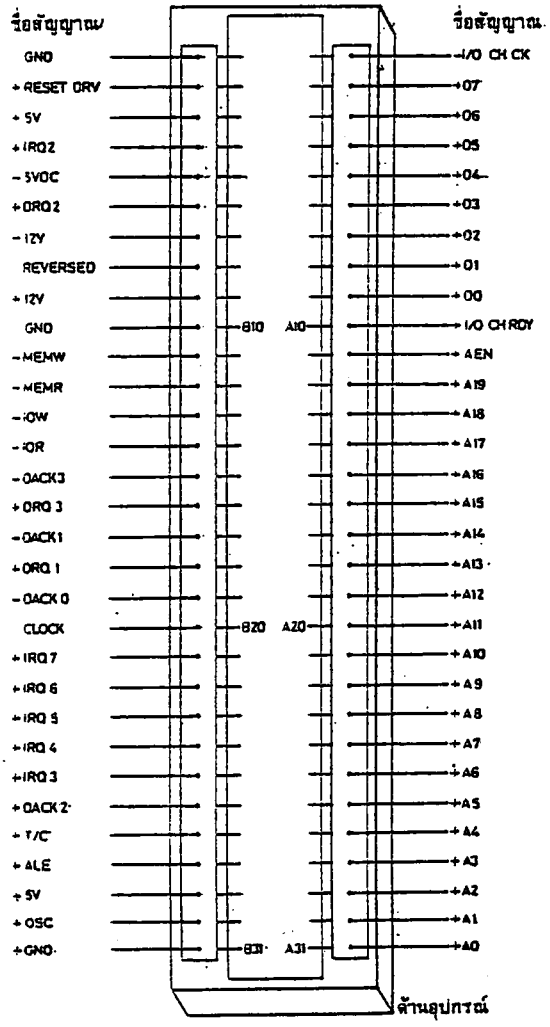
บทที่ 4

การเชื่อมต่อกับไมโครคอมพิวเตอร์ไอบีเอ็มพีซี

(IBM/PC Interfacing)

4.1 การเชื่อมต่อไมโครคอมพิวเตอร์

ในบทนี้จะได้อธิบายหลักการเชื่อมต่อระหว่างสัญญาณของตัวตรวจวัดที่เป็นสัญญาณอนาลอกมาเป็นสัญญาณดิจิตอล และเปลี่ยนจากดิจิตอลไปเป็นอนาลอกอีกด้วยเพื่อนำไปใช้งานต่อไป



รูปที่ 4.1 สัญญาณอินเตอร์เฟสกับระบบไมโครคอมพิวเตอร์

IBM PC ได้ทำการออกแบบให้สามารถเพิ่มเติมวงจรอินเตอร์เฟส (Interface circuit) เข้าไปในภายหลังได้ โดยผ่านทางสล็อตที่อยู่บนเมนบอร์ดซึ่งมีจำนวน 5 สล็อต แต่ละสล็อตมีจำนวนขาทั้งหมด 62 ขาแบ่งออกเป็นสองด้าน ๆ ละ 31 ขาดังรูปที่ 4.1 แต่ละขาของสล็อตเหล่านี้จะต่อเชื่อมกับขาสัญญาณต่าง ๆ บนเมนบอร์ดทำให้การสร้างวงจรอิน

เตอร์เฟสกับ IBM PC สามารถทำได้สะดวกซึ่งเส้นสัญญาณที่เชื่อมต่อกับสล็อตเหล่านี้จะประกอบไปด้วยเส้นสัญญาณของบัสแอดเดรส (Address bus), บัสข้อมูล (Data bus), บัสควบคุมสำหรับการอ่าน/เขียนข้อมูลจากหน่วยความจำหรือจากพอร์ท I/O, เส้นสัญญาณสำหรับการขออินเตอร์รัพท์ของวงจรอินเตอร์เฟส, เส้นสัญญาณสำหรับการขอติดต่อกับหน่วยความจำโดยไม่ผ่านซีพียู (Data Memory Access ;DMA), สัญญาณฐานเวลา (Timing Signal) ต่าง ๆ ที่ใช้ในระบบ, เส้นสัญญาณแสดงการขอรีเฟรชหน่วยความจำและสัญญาณสำหรับตรวจสอบความผิดพลาด (I/O Check)

นอกจากเส้นสัญญาณเหล่านี้แล้ว สล็อตบนเมนบอร์ดยังเชื่อมต่อกับแหล่งจ่ายไฟต่าง ๆ ที่ใช้ในระบบด้วยคือ +5Vdc ,-5Vdc ,+12Vdc และ -12Vdc

การแบ่งแอดเดรสของ IBM/PC

ใน IBM/PC มีการจัดสรรการจัดแอดเดรสต่าง ๆ ไว้แน่นอน ซึ่งได้แสดงดังตารางที่ 4.1 ส่วนที่เรานำมาใช้ในการ Decode address อยู่ในส่วนของ Prototype Card ซึ่งมีแอดเดรสอยู่ในช่วง 300-31F ทั้งหมด 32 พอร์ทแอดเดรส

ตารางที่ 4.1 การแบ่งแอดเดรสของ IBM/PC

Hex range	Usage	
000-00F	DMA chip 8237A-5	
020-021	Interrupt 8259A	
040-043	Timer 8253-5	Assigned to
060-063	PPI 8255A-5	system board
080-083	DMA page registers	components
0Ax	NMI mask register	
0Cx	Reserved	
0Ex	Reserved	
100-1FF	Not usable	
200-20F	Game control	

210-217	Expansion unit	
220-24F	Reserved	
278-27F	Reserved	
2F0-2F7	Reserved	Assigned to
2F8-2FF	Asynchonous communication(2)	feature
300-31F	Prototype card	card ports
320-32F	Fixed disk	
378-37F	Printer	
380-38C	SDLC communication	
380-389	Binary synchonous communication(2)	
3A0-3A9	Binary synchonous communication(1)	
380-38F	IBM monochrome display printer	
3C0-3CF	Reserved	
3D0-3DF	Colour/graphics	
3E0-3E7	Reserved	
3F0-3F7	Diskette	
3F8-3FF	Asynchonous communications(1)	

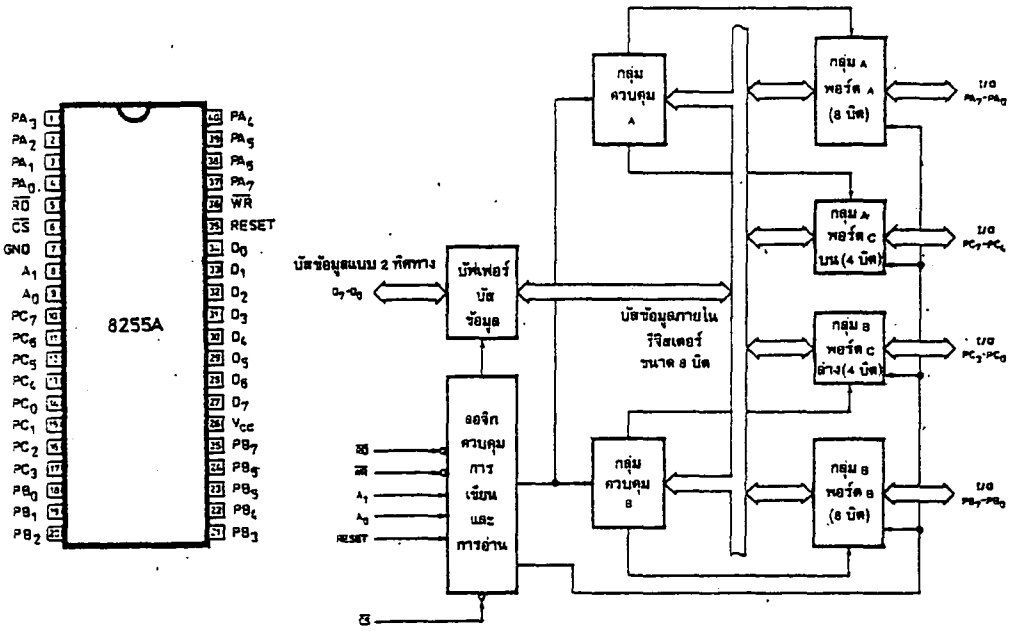
4.2 วงจรรับส่งข้อมูลแบบขนาน (8255 PPI)

เมื่อเราทราบรายละเอียดภายในของ IBM/PC บ้างแล้วว่ามีสล็อตอะไรบ้าง และมีการจัดแบ่งแอดเดรสไว้อย่างไร ดังนั้นเราจะนำไปสู่การเชื่อมต่อ IBM/PC กับ 8255 ดังนี้

การ Decode address

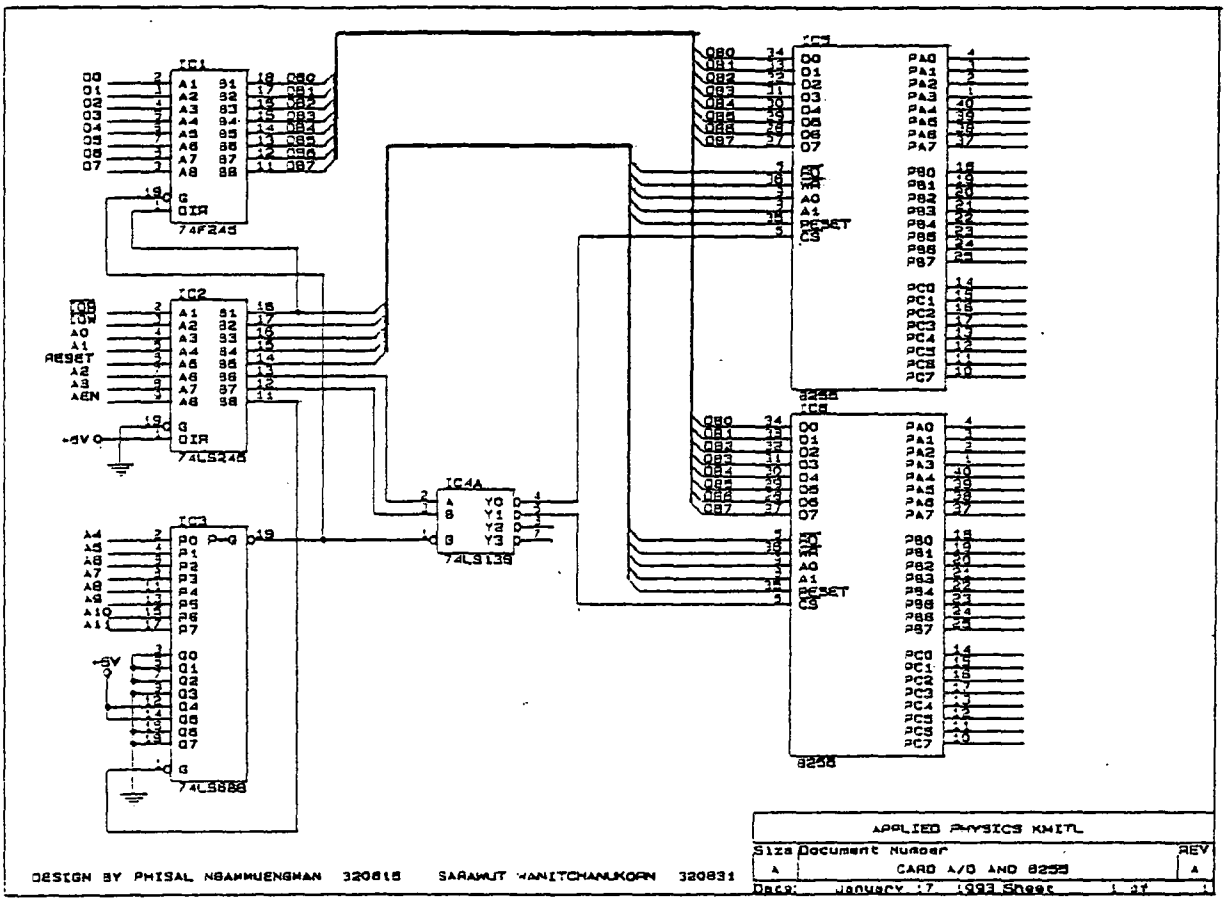
ในโครงงานนี้ทำการ Decode แอดเดรส 300H ถึง 307H นำมาใช้ในการเชื่อมต่อกับ 8255PPI ซึ่งแอดเดรสเหล่านี้อยู่ในส่วนของ Prototype card

หลังจากเราได้วงจร Decode เสร็จเรียบร้อยแล้วจึงนำวงจร Decode นี้มาเชื่อมต่อกับ 8255 แต่ก่อนที่เราจะทำการเชื่อมต่อกับ 8255 นั้น จะต้องรู้ถึงโครงสร้างของมันคร่าว ๆ ดังนี้คือ 8255 เป็นไอซีที่มี 40 ขา ได้รับการออกแบบให้มีสัญญาณเพื่อเชื่อมต่อกับ IBM/PC ซึ่งตัว 8255 เองนั้นเป็นไอซีที่ต่อเป็นพอร์ทให้ไมโครโปรเซสเซอร์ได้ 3 พอร์ท คือ Port A, Port B และ Port C สามารถแบ่งได้ออกเป็น 2 ส่วนคือ พอร์ท c ล่าง และ พอร์ท c บน ที่พิเศษคือ พอร์ททุกพอร์ทเป็นไปได้ทั้งอินพุทและเอาท์พุท



รูปที่ 4.2 แผนผังวงจรภายในและการจัดขาของไอซี 8255

เมื่อทราบลักษณะโครงสร้างภายในแล้วจึงนำเอาวงจร Decode มาเชื่อมต่อกันเข้ากับ 8255 ดังแสดงวงจรดังรูปที่ 4.3



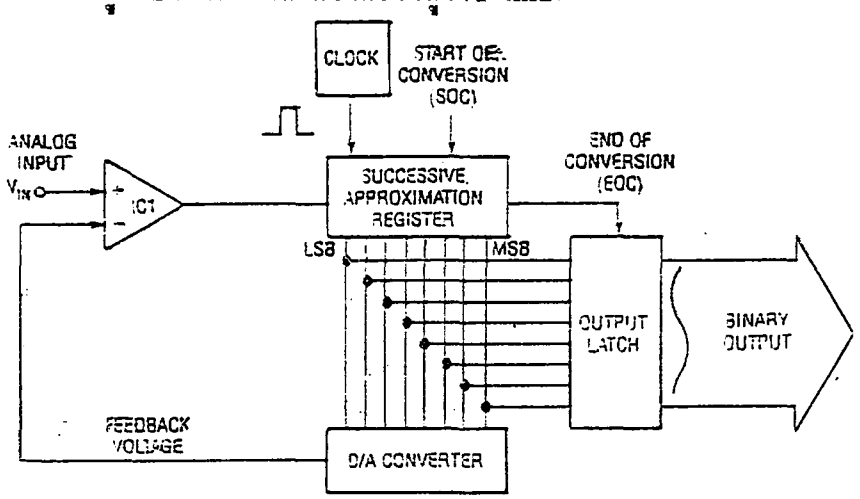
รูปที่ 4.3 การต่อ 8255 กับวงจร Decode

ซึ่งวงจรดังกล่าวจะมีการใช้เรียกพอร์ทของ 8255 ดังนี้

		A11.....A0		
พอร์ท A	300H	0011	0000	0000
พอร์ท B	301H	0011	0000	0001
พอร์ท C	302H	0011	0000	0010
คอนโทรลพอร์ท	303H	0011	0000	0011
พอร์ท A1	303H	0011	0000	0011
พอร์ท B1	305H	0011	0000	0100
พอร์ท C1	306H	0011	0000	0110
คอนโทรลพอร์ท	307H	0011	0000	0111

4.3 วงจรแปลงสัญญาณอนาลอกเป็นดิจิตอล (Analog To Digital Convertor,A/D)

เนื่องจากสัญญาณเอาท์พุทที่ได้จากเลเซอร์ดิสเพลซเมนต์เซ็นเซอร์ (LDS) เป็นสัญญาณอนาลอก ดังนั้นจึงต้องทำการแปลงเป็นสัญญาณดิจิตอลเพื่อส่งเข้าไปเก็บในหน่วยความจำของไมโครคอมพิวเตอร์ได้ สำหรับโครงงานนี้ได้เลือกใช้ไอซี A/D เบอร์ ADC 574AJ ชนิด 12 บิทซึ่งเป็นไอซีชนิดที่คอมแพทติเบิล (Compatible) กับไมโครโปรเซสเซอร์ ซึ่งมีการทำงานเป็นแบบประมาณค่าทีละบิท (Successive approximation) โดยได้เลือกใช้ค่าแอดเดรสสำหรับการดีโคด (Decode) คือ 308H-309H ADC ชนิดนี้มีการใช้งานกันอย่างกว้างขวางโดยเฉพาะอย่างยิ่งเกี่ยวกับการอินเตอร์เฟสกับคอมพิวเตอร์ เนื่องจากมันมีทั้งความละเอียดสูง(12 บิท) และความเร็วสูง(1 MHz)



รูปที่ 4.4 บล็อกไดอะแกรมของ A/D Convertor แบบบันทึกการประมาณค่าอย่างต่อเนื่องแบบ SAR

4.3.1 A/D แบบการประมาณค่าผลสำเร็จ (Successive Approximations,SA)

เทคนิคการแปลงชนิดนี้อาศัยหลักการการเปรียบเทียบระหว่างสัญญาณอินพุทที่ไม่ทราบค่ากับแรงดันหรือกระแสจากการแปลงสัญญาณดิจิตอลให้เป็นสัญญาณอนาลอก (Digital To Analog Convertor, DAC) ชนิดความแม่นยำสูง ซึ่งอินพุทของ DAC ก็คือค่าดิจิตอลที่เอาท์พุทของ ADC นั่นเอง หัวใจของ SA Convertor คือ อุปกรณ์ที่เรียกว่า รีจีสเตอร์ประมาณค่าแบบผลสำเร็จ (Successive approximation register :SAR) ซึ่งเป็นอุปกรณ์ที่มีจุดประสงค์ต่างจากวงจรนับทั่ว ๆ ไปอย่างมากดังแสดงในรูปที่ 4.4 การแปลงเริ่มต้นเมื่อสัญญาณอนาลอกถูกป้อนให้กับคอนเวอร์เตอร์ และพัลส์ของการคอนเวอร์เตอร์ได้เริ่มขึ้นถูกป้อนให้กับตัว SAR พัลส์สัญญาณนาฬิกาลูกแรกที่ป้อนให้กับตัว SAR จะ"ON" เอาท์พุทของบิทนับสูงสุด ดังนั้นจึงเป็นการปรับให้เอาท์พุทของ DAC เป็น 50% ของแรงดันเอาท์พุทของอินเวอร์เตอร์ ตัว SAR จะมองไปยังเอาท์พุทของวงจรเปรียบเทียบว่าเอาท์พุทของ

DAC มีค่ามากกว่าหรือน้อยกว่าสัญญาณอนาลอกทางอินพุท ถ้าแรงดันของ DAC มีค่ามากกว่าวงจรเปรียบเทียบจะยังคงอยู่ในสภาวะ "OFF" ดังนั้นตัว SAR จะ "OFF" บิทนัยสำคัญสูงสุดลงและให้ชื่อว่าสภาวะ "0" ถ้าแรงดันของ DAC มีค่าน้อยกว่าสัญญาณอนาลอกทางอินพุทวงจรเปรียบเทียบจะยังคงทำงานอยู่ ดังนั้นตัว SAR จะยังคงปล่อยให้บิทนัยสำคัญสูงสุด "ON"อยู่และเราเรียกสภาวะนี้ว่า "1" ซึ่งสภาวะทั้งสองนี้จะกระทำภายในพัลส์ของสัญญาณนาฬิกาเพียงพัลส์เดียว บนสัญญาณนาฬิกาถัดไป ตัว SAR จะ "ON" บิทนัยสำคัญสูงสุดอันดับสองและทำการตรวจสอบผลลัพธ์ที่ได้อีกครั้งหนึ่งจากวงจรเปรียบเทียบ ถ้าสัญญาณจาก DAC ครั้งใหม่มีค่ามากกว่าแรงดันอินพุท เอาท์พุทของวงจรเปรียบเทียบจะยังไม่มี ดังนั้นตัว SAR จะ "OFF" บิทนัยสำคัญสูงสุดอันดับสองและเรียกมันว่า "0" แต่ถ้าสัญญาณจาก DAC มีค่าน้อยกว่าวงจรเปรียบเทียบจะยังคงทำงาน และตัว SAR จะปล่อยให้บิทนัยสำคัญสูงสุดอันดับสองนี้ "ON"

ตัว SAR จะพิจารณาแต่ละบิทด้วยวิธีเดียวกัน(บิทนัยสำคัญสูงสุดถึงต่ำสุด) จนกระทั่งทุกบิทถูกพิจารณาหมด เนื่องจาก 1 บิทถูกหาค่าภายใน 1 พัลส์ - ADC ขนาด 8 บิทจึงใช้สัญญาณนาฬิกาเพียง 8 พัลส์ก็สามารถทำการแปลงได้จนจบกระบวนการ เมื่อบิทนัยสำคัญต่ำสุดถูกพิจารณาเสร็จสิ้นแล้ว ตัว SAR จะส่งสัญญาณสิ้นสุดการแปลงไปทำการค้างผลลัพธ์ที่ได้ซึ่งเป็นเลขฐานสองทางเอาท์พุทไว้

SA Convertorเป็นตัวแปลงสัญญาณที่มีประสิทธิภาพสูงมากและสามารถทำการแปลงสัญญาณได้รวดเร็วพร้อมทั้งมีความละเอียดสูงอีกด้วย คอนเวอร์เตอร์หลายตัวที่ใช้เทคนิคนี้ทำการแปลงสัญญาณอนาลอกเป็นดิจิตอล 12 บิทได้โดยใช้เวลาน้อยกว่า 10 ไมโครวินาที

4.3.2 คุณสมบัติของ ADC574AJ กับการวัดของระบบเลเซอร์ดิสเพลซเมนต์เซ็นเซอร์

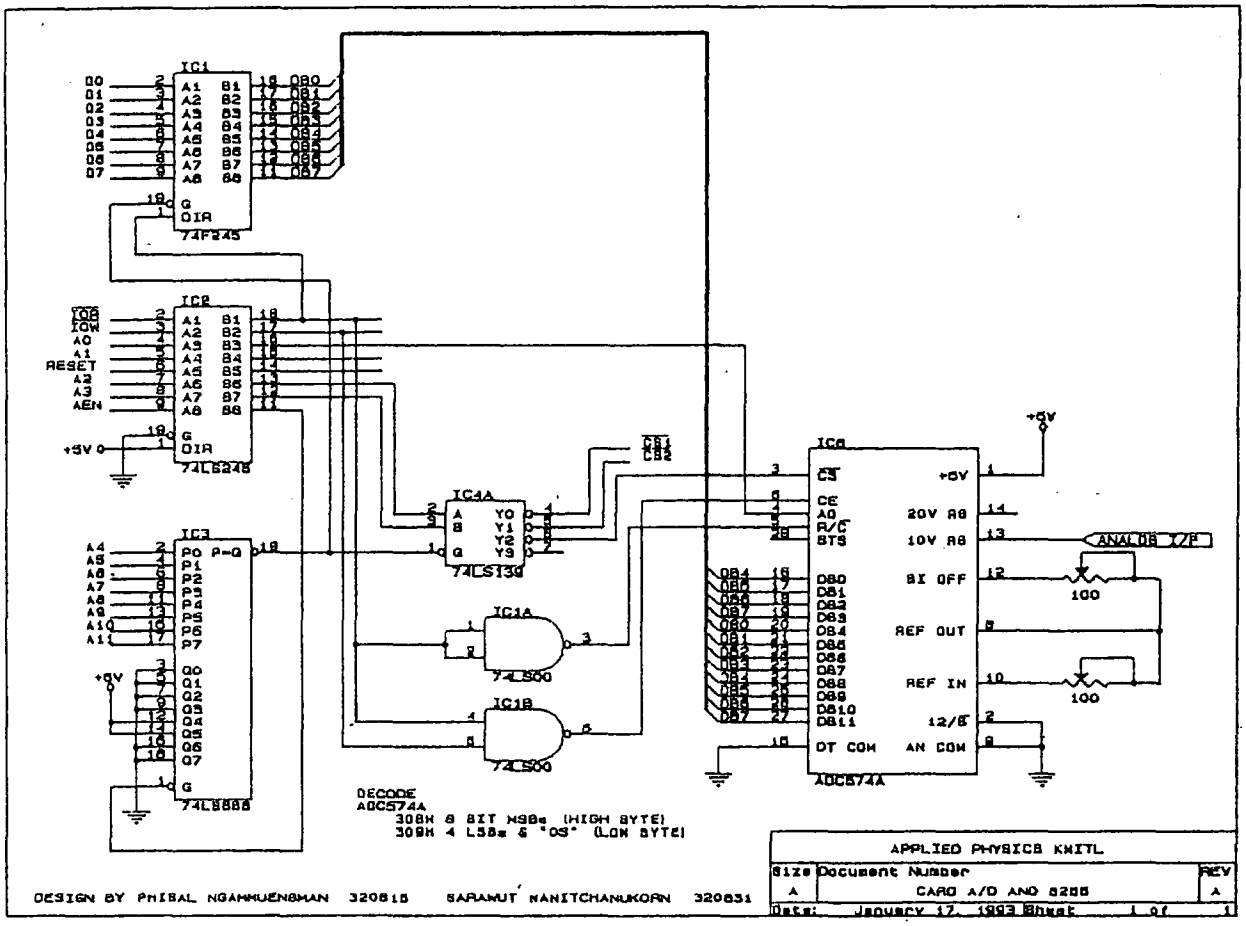
- อัตราการแปลงข้อมูล (Conversion time) เท่ากับ 25 ไมโครวินาที (40 KHz)พบว่า ADC574AJ สามารถทำการแปลงข้อมูลได้ทันกับความเร็วของเลเซอร์ดิสเพลซเมนต์เซ็นเซอร์ในโหมดการตอบสนองต่อความเร็วสูงสุดของการวัด คือ 0.7 มิลลิวินาที (700 Hz)

- ความต่างศักย์ของ ADC574AJ ที่รับได้โดยตรง +5 โวลต์ถึง -5 โวลต์ โดยเลเซอร์ดิสเพลซเมนต์เซ็นเซอร์มีค่าอนาลอกเอาท์พุทอยู่ในช่วง +4 โวลต์ถึง -4 โวลต์

- ความละเอียดของข้อมูลเท่ากับ 12 บิท ดังนั้นสัญญาณอินพุทจึงถูกแบ่งเป็น 4096 ระดับ ระดับละ 2.44 mv

- การติดต่อกับ Data bus ของระบบคอมพิวเตอร์เลือกใช้ค่าแอดเดรสสำหรับการ

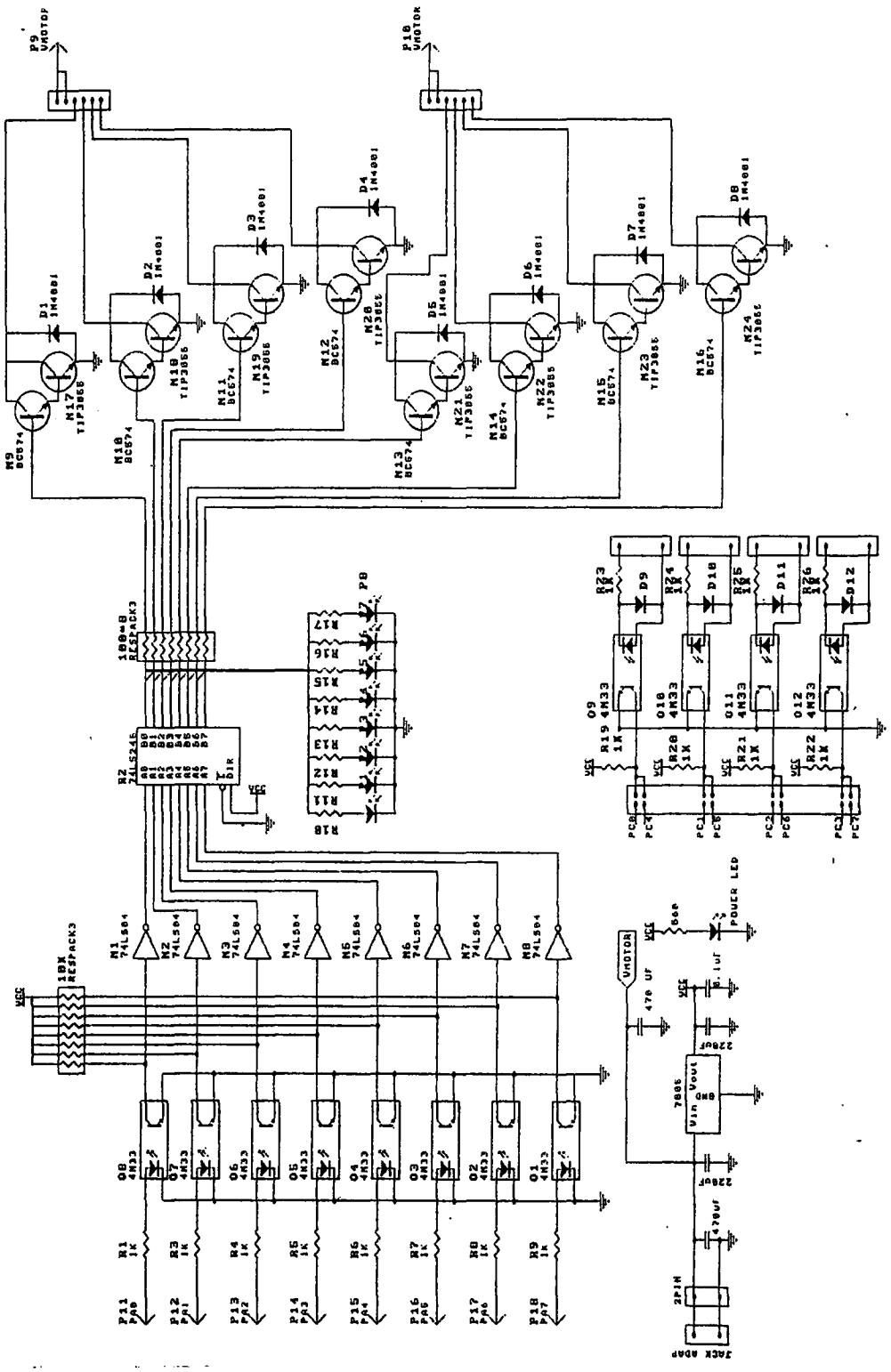
ดีโคด (Decode) คือ 308H เป็นการอ่านค่า 8 บิทบน (Most significant bit) จาก ADC และ 309H เป็นการอ่านค่า 4 บิทล่าง (Least significant bit)



รูปที่ 4.5 แสดงวงจรเชื่อมต่อส่วนของวงจรแปลงสัญญาณอนาลอกเป็นดิจิตอล

4.4 การเชื่อมต่อวงจรรับส่งข้อมูลแบบขนาน (8255PPI) กับวงจรขับกำลังสเตปปิ้งมอเตอร์

วงจรขับกำลังสเตปปิ้งมอเตอร์เป็นส่วนรับข้อมูลจาก 8255PPI เพื่อการขับหมุนของสเตปปิ้งมอเตอร์ในการควบคุมตำแหน่งและทิศทางของแท่นสแกน X-Y ซึ่งเลือกวงจรที่ใช้ทรานซิสเตอร์แบบเอ็นพีเอ็นต่อกันในลักษณะดาร์ลิงตัน (Darlington) เพื่อให้ได้กระแสที่สูงพอในการขับสเตปปิ้งมอเตอร์



รูปที่ 4.6 วงจรขับกำลังสเตปปิ้งมอเตอร์

- Port A ใช้เป็นตัวส่งข้อมูลไปควบคุมการเคลื่อนที่ของแกน X และแกน Y โดยมีแอดเดรสที่ 300H
- ส่วน 8255PPI อีกตัวนั้นสำรองสำหรับการขยายระบบในอนาคต

การต่อ 8255 กับวงจรขับกำลังใช้วิธีการต่อขาข้อมูลของพอร์ทต่าง ๆ เข้ากับส่วนรับสัญญาณชนิดออปโต้ (Opto-isolated) จำนวน 8 ช่อง

บทที่ 5

โครงสร้างของโปรแกรม

จากโครงสร้างของโปรแกรมสามารถแบ่งโปรแกรมเป็น 2 ส่วนดังนี้

5.1 โปรแกรมควบคุมระบบฮาร์ดแวร์

5.1.1 ส่วนควบคุม XY-Table

เป็นโปรแกรมส่วนที่ใช้ในการควบคุมการเคลื่อนที่ของ XY-Table ใช้ในการควบคุมตำแหน่งของชิ้นงานบน XY-Table ให้เคลื่อนที่ไปยังตำแหน่งที่เหมาะสมกับขนาดของวัตถุที่ต้องการที่จะทำการสแกนก่อนเริ่มทำการสแกนพื้นผิว

5.1.2 ส่วนการสแกนและเก็บข้อมูล

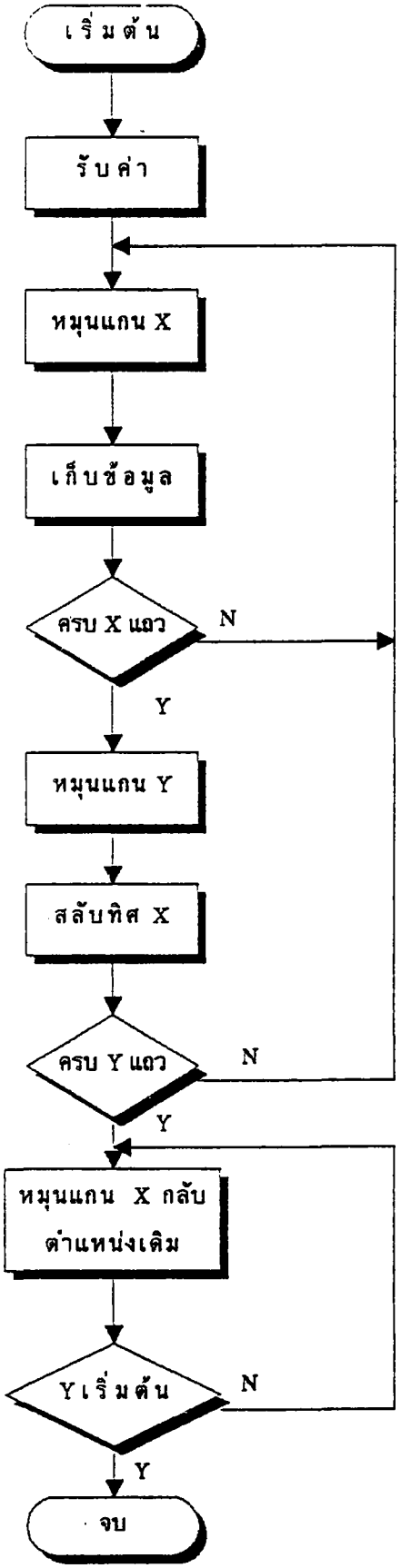
เป็นโปรแกรมที่ใช้ในการสแกนพื้นผิววัตถุ โดยจะรับค่าจาก A/D แล้วนำไปเก็บในหน่วยความจำ จากนั้นจะส่งข้อมูลไปควบคุม XY-Table ไปยังตำแหน่งต่อไปที่จะทำการเก็บข้อมูลจนครบทั่วทั้งบริเวณพื้นผิวที่ต้องการ การสแกนจะสแกนพื้นผิวในลักษณะเป็นรูปสี่เหลี่ยมจัตุรัสและเริ่มสแกนจากมุมบนซ้ายของจุดที่เริ่มสแกนไปทางขวา และสแกนไปกลับจนถึงจุดสิ้นสุดของการสแกน (มุมล่างขวา) โดยขนาดของวัตถุจะต้องมีขนาดไม่เกินช่วงที่ XY-Table สามารถทำการสแกนได้ในที่นี้กำหนดเป็น 10 * 10 ซม. ส่วนระยะห่างระหว่างจุดที่ทำการวัดสามารถวัดได้โดยอยู่ในช่วง 5 μm ถึง 1000 μm แสดงการเก็บข้อมูลเป็นโฟลว์ชาร์ทได้ดังรูปที่ 5.1

5.2 โปรแกรมการสร้างภาพและการแสดงผล

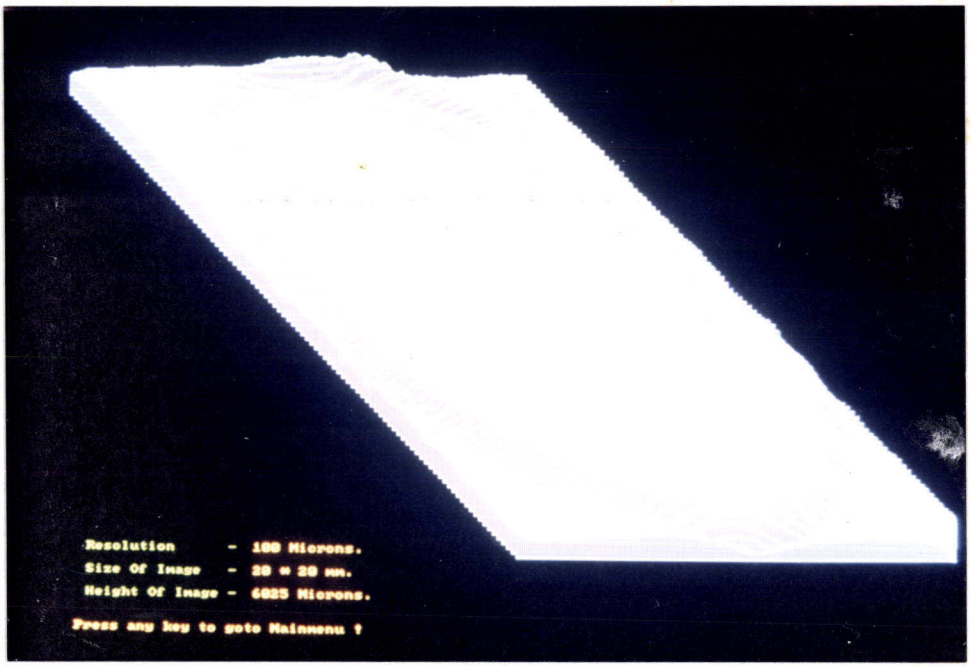
เป็นการนำเอาข้อมูลที่ได้จากการสแกนพื้นผิววัตถุมาแสดงผลทางจอคอมพิวเตอร์ โดยใช้การโปรแกรมภาษาปาสคาลทำงานบนเครื่อง 486DX-33 ความละเอียดของจอภาพในการแสดงผลแสดงในโหมด 800 * 600 จุด สามารถแสดงจำนวนสีได้สูงสุดในหนึ่งหน้าจอ 256 สี แบ่งการแสดงผลเป็น 3 แบบดังนี้

5.2.1 การแสดงผลโดยการจำลองภาพ 3 มิติแบบธรรมดา

เป็นการแสดงระดับความสูงต่ำของภาพโดยข้อมูลที่มีค่ามากจะมีระดับความสูงสูงกว่าข้อมูลที่มีค่าน้อยกว่า สามารถเลือกขนาดของภาพได้ 3 ขนาดและแบ่งระดับของข้อมูลได้ในระดับต่าง ๆ แสดงดังรูปที่ 5.2



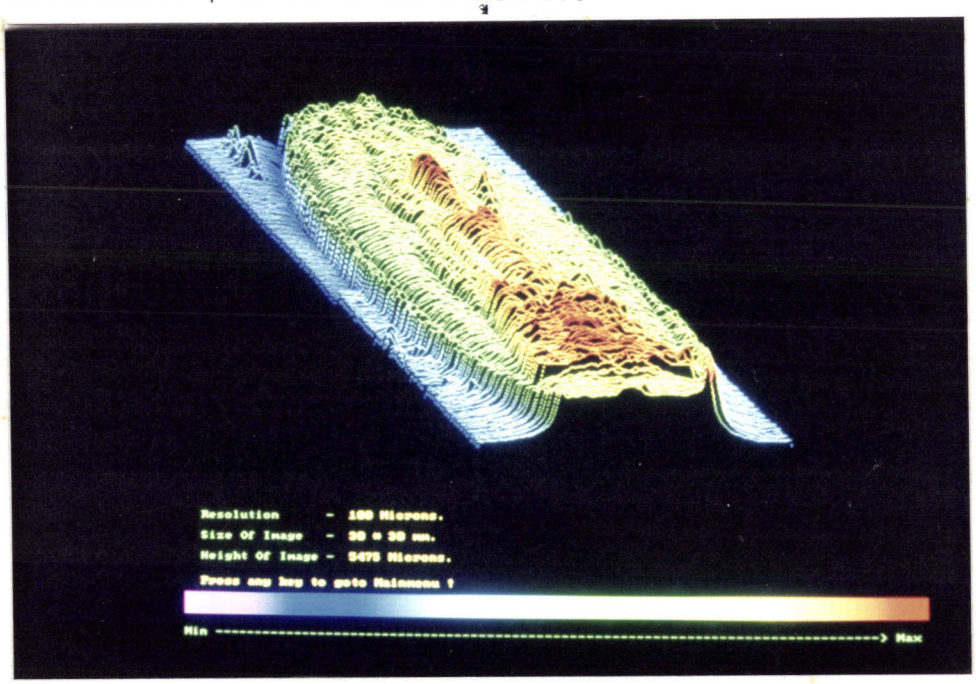
รูปที่ 5.1 แสดงโฟลว์ชาร์ทวิธีการเก็บข้อมูลที่ได้จากการสแกนในระนาบ X-Y



รูปที่ 5.2 แสดงภาพการแสดงพื้นผิวโดยการจำลองภาพ 3 มิติแบบธรรมดา

5.2.2 การแสดงผลโดยการจำลองภาพแบบให้เฉดสีแสดงความสูงต่ำ

เป็นการแสดงระดับความสูงต่ำของวัตถุเหมือนวิธีแรก แต่ระดับความสูงต่ำของภาพแทนด้วยสีเฉดต่าง ๆ ได้ 240 สี แสดงดังรูปที่ 5.3

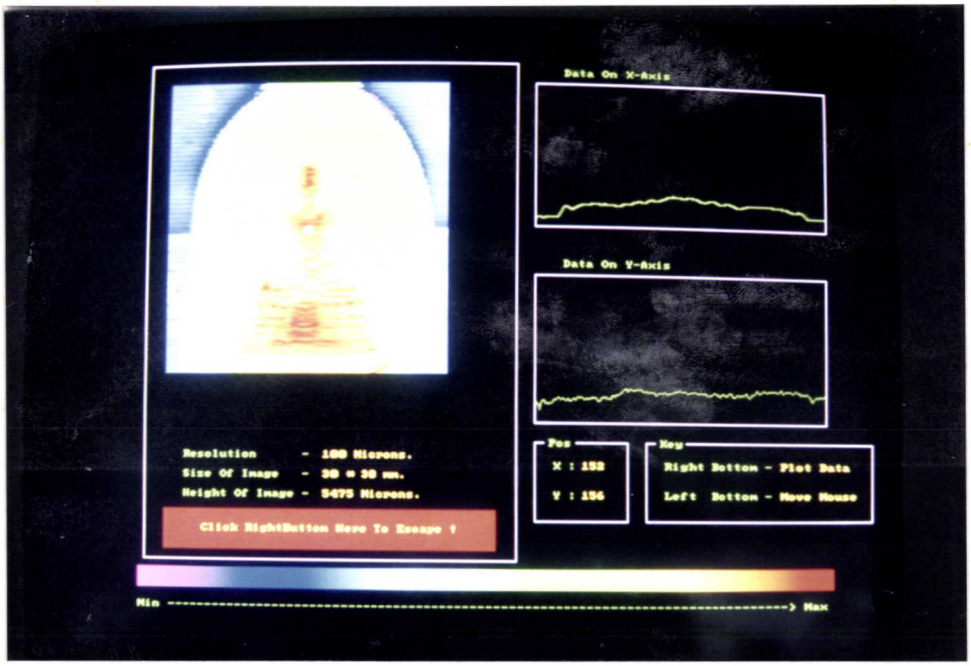


รูปที่ 5.3 แสดงภาพการแสดงพื้นผิวโดยการจำลองภาพแบบให้เฉดสีแสดงความสูงต่ำ

5.2.3 การแสดงผลโดยการวิเคราะห์ภาคัดขวางของพื้นผิววัตถุ

เป็นการแสดงพื้นผิวของวัตถุจากมุมมองด้านบน (Top view) โดยใช้เฉดสีต่าง ๆ

240 สีแสดงระดับความสูงต่ำของภาพเหมือนวิธีที่ 2 และสามารถแสดงข้อมูลของพื้นผิววัตถุในระนาบ X และ Y โดยการใช้เม้าส์คลิ๊กไปที่ตำแหน่งใด ๆ ของภาพซึ่งมีการแสดงตำแหน่งบนภาพไว้ด้วย แสดงดังรูปที่ 5.4



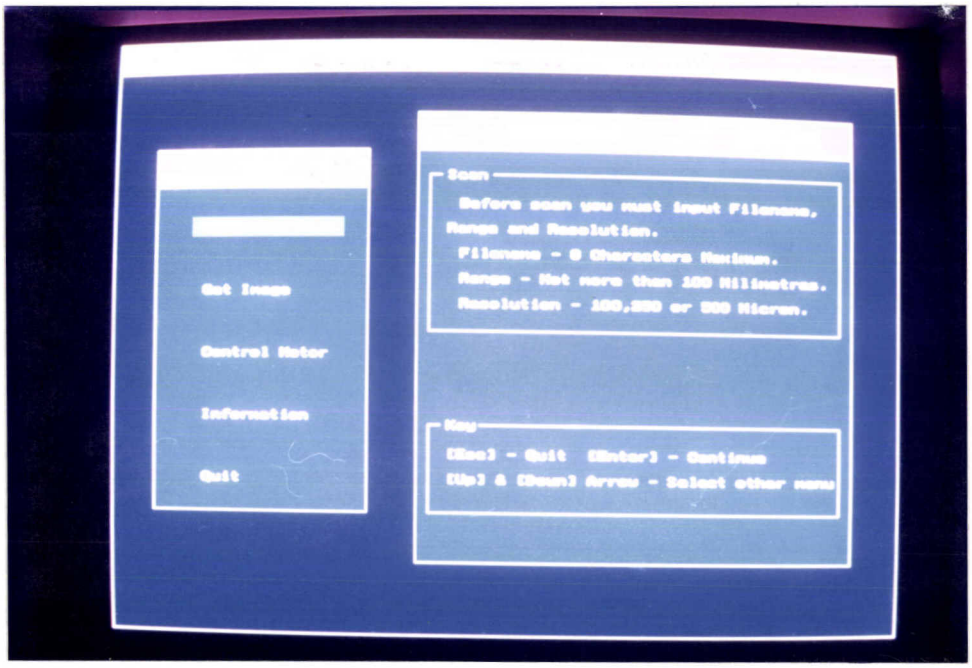
รูปที่ 5.4 แสดงภาพการแสดงพื้นผิววัตถุโดยการวิเคราะห์ภาคตัดขวาง

5.3 การใช้งานโปรแกรม Laser Surface Scanning System

1. เข้าโปรแกรมโดยการเรียก Lsss หน้าจอจะแสดงดังรูปที่ 5.5

รูปที่ 5.5 แสดงภาพหน้าจอหลังจากเรียกโปรแกรมแล้ว

2. กดคีย์ใด ๆ หน้าจอจะแสดงเมนูหลัก (Main Menu) ดังรูปที่ 5.6



รูปที่ 5.6 แสดงภาพเมนูหลัก (Main Menu)

3. เมนูหลักประกอบไปด้วยเมนูย่อย 5 เมนูด้วยกันคือ

3.1 Scan

3.2 Get Image

3.3 Control Motor

3.4 Information

3.5 Quit

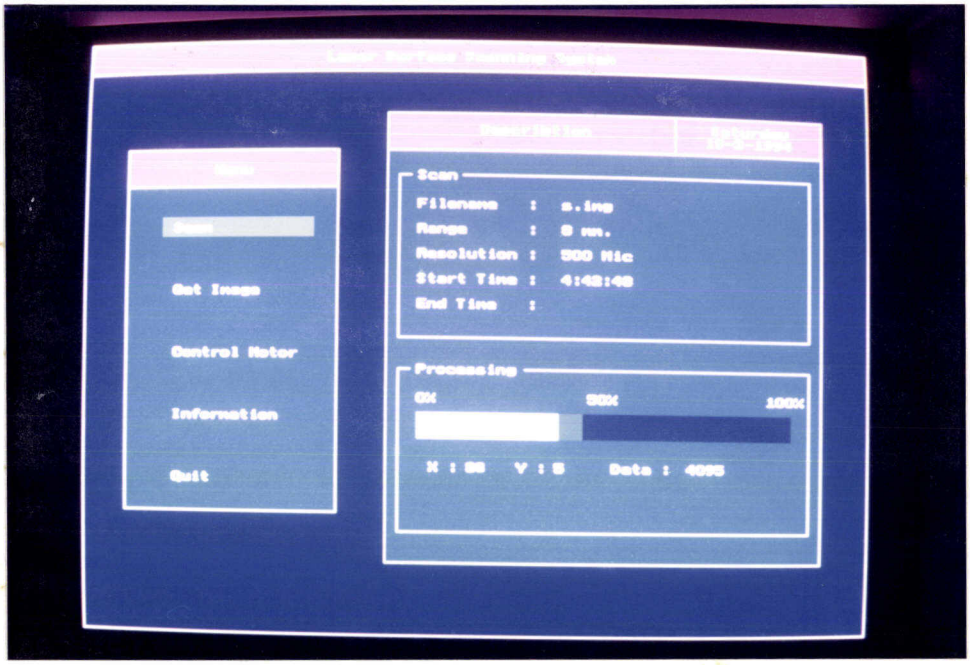
เราสามารถเลือกเมนูย่อยใด ๆโดยการเลื่อนแถบสว่างไปที่เมนูย่อยนั้น ๆและกดคีย์ [Enter] การใช้คีย์ในเมนูย่อยดูได้จากช่อง Key ซึ่งแสดงไว้แล้วสามารถอธิบายเมนูย่อยเป็นส่วน ๆ ดังนี้

1. Scan

ใช้ในการเก็บข้อมูลเป็นไฟล์ (File) โดยต้องมีการกำหนด

- ชื่อไฟล์ (Filename) มีความยาวได้ไม่เกิน 8 ตัวอักษร ไฟล์ที่เก็บจะมีนามสกุล .img
- ระยะในการสแกน (Range) มีระยะไม่เกิน 100 มิลลิเมตร
- ความละเอียดในการสแกน (Resolution) สามารถเลือกได้ 3 ระดับ คือ 100 250 หรือ 500 ไมครอน

ในระหว่างสแกนเก็บข้อมูลผู้ใช้สามารถกดคีย์ใด ๆ เพื่อเลิกกลางคันได้ ซึ่งจะมีข้อความขึ้นให้ยืนยัน (Yes/No) ถ้าผู้ใช้กดคีย์ Y ก็จะเลิกการสแกนแต่ถ้ากดคีย์ N ก็จะทำการสแกนเก็บข้อมูลต่อไปจนสิ้นสุด แสดงภาพหน้าจอการสแกนเก็บข้อมูลดังรูปที่ 5.7



รูปที่ 5.7 แสดงภาพหน้าจอการสแกนเก็บข้อมูล

2. Get Image

เป็นส่วนแสดงผลเป็นภาพโดยต้องมีการกำหนด

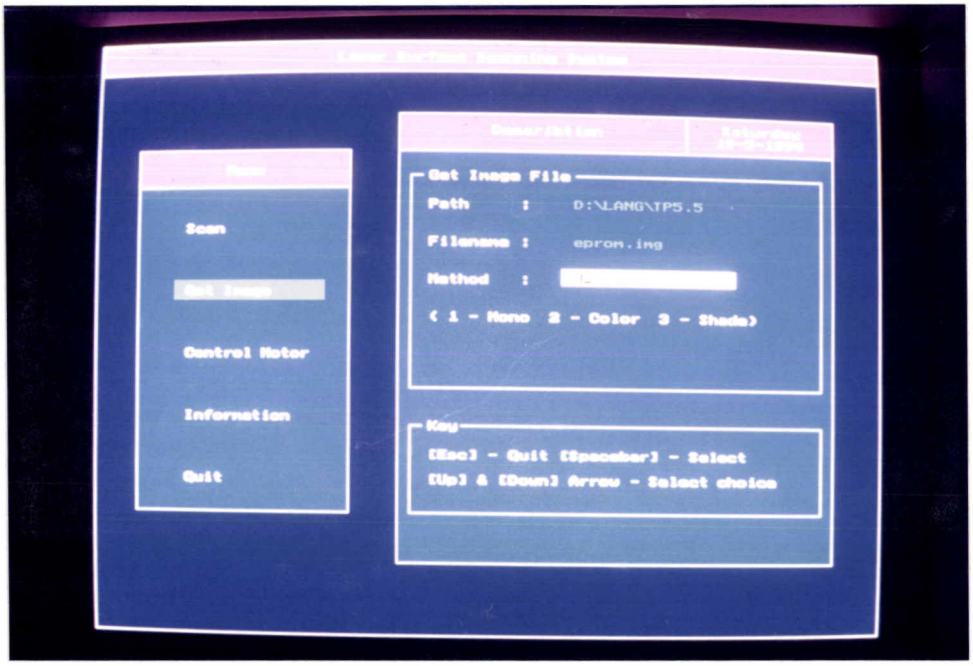
- Path ที่เก็บข้อมูล โดยเริ่มต้นจะกำหนดเป็น Path ที่อยู่ปัจจุบัน
- ชื่อไฟล์ (Filename) ของข้อมูล (ไม่ต้องใส่นามสกุล)
- วิธีในการแสดงผล (Method) มี 3 วิธีด้วยกันคือ
  1. Mono - แสดงระดับความสูงต่ำของภาพโดยมีสีเดียว
  2. Color - เหมือนการแสดงผลวิธีที่ 1 แต่ระดับความสูงต่ำของภาพแทนด้วยสีต่าง ๆ ได้ 240 สี
  3. Shade - แสดงระดับความสูงต่ำของภาพโดยใช้สี 240 สีในมุมมอง Top View และสามารถใช้ Mouse คลิ๊กที่ตำแหน่งใด ๆ ของภาพเพื่อ แสดงค่าของข้อมูลในแนวแกน X และแกน Y

วิธีที่ 1 และ 2 ต้องมีการกำหนด

- จุดเริ่มต้นภาพในแนวแกน Y อยู่ระหว่าง 1-200
- ขนาดภาพ (Size) เลือกได้ 3 ขนาดคือ 1,2 และ 3

- ระดับความสูงของข้อมูล (Levels) สามารถเลือกได้ระหว่าง 1-100

แสดงภาพหน้าจอการนำข้อมูลมาทำการแสดงผลได้ดังรูปที่ 5.8



รูปที่ 5.8 แสดงภาพหน้าจอการนำข้อมูลมาทำการแสดงผล

3. Control Motor

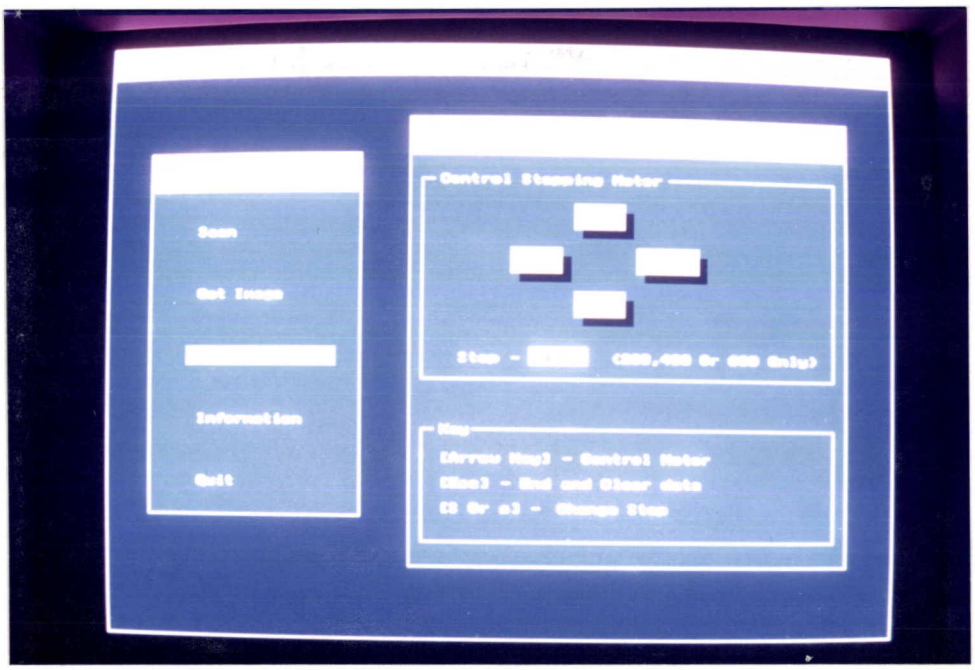
ใช้ในการควบคุม Stepping Motor ซึ่งใช้ในการกำหนดตำแหน่งเริ่มต้นของวัตถุที่ต้องการสแกนไปในตำแหน่งต่าง ๆ ของ X-Y Table สามารถเปลี่ยนสเตปการหมุนของมอเตอร์ได้โดยการกดคีย์ [S] และใส่ค่าของสเตปที่ต้องการเข้าไป (สามารถเลือกสเตปการหมุนของมอเตอร์ได้ 3 ระดับคือ 200,400 และ 600) หากไม่มีการเปลี่ยนค่าสเตปก็จะมีค่า 400 แสดงหน้าจอการควบคุมมอเตอร์ได้ดังรูปที่ 5.9

4. Information

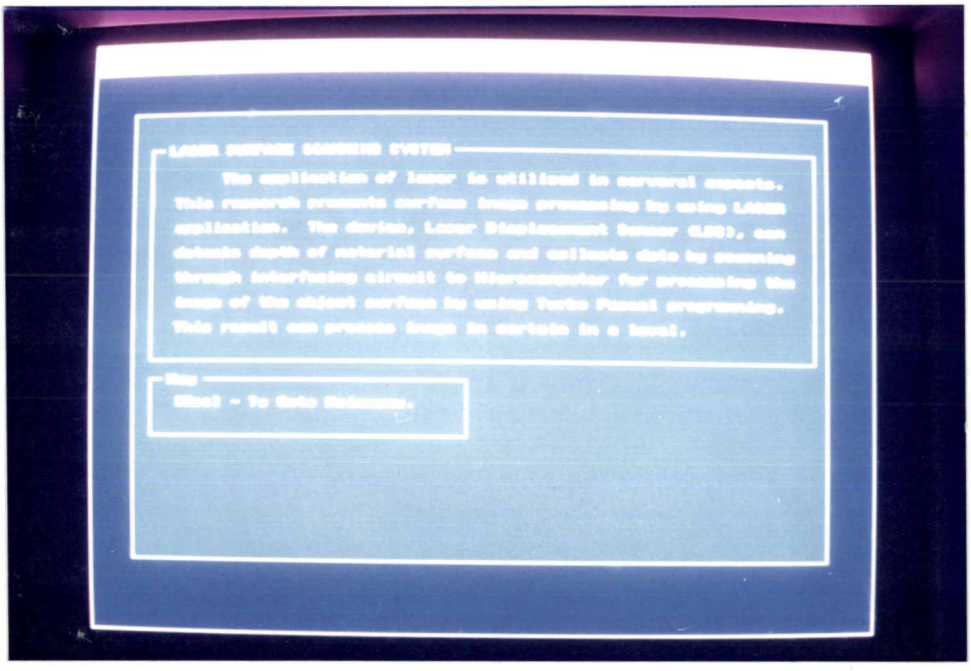
ส่วนนี้เป็นส่วนที่ให้ข้อมูลเกี่ยวกับโครงงานโดยสังเขป สามารถออกไปเมนูหลักโดยการกดคีย์ [Esc] แสดงหน้าจอของส่วน Information ได้ดังรูปที่ 5.10

5. Quit

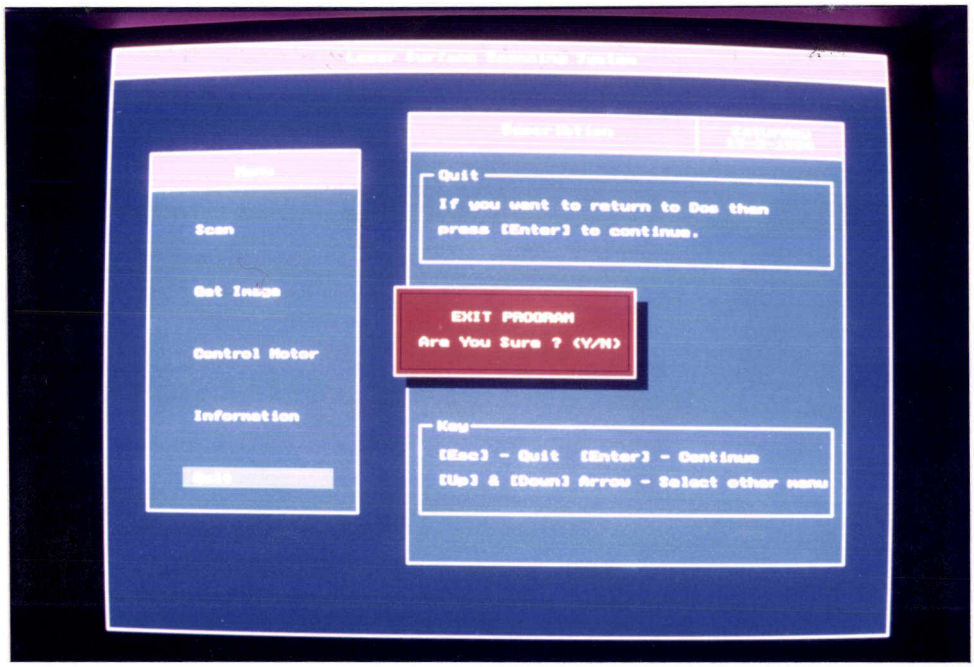
ใช้ในการออกจากโปรแกรมไปสู่ Dos หลังจากเลือกเมนูนี้แล้วโปรแกรมจะให้ผู้ใช้ยืนยันอีกครั้งโดยให้เลือก (Yes/No) ถ้าผู้ใช้กด Y ก็จะออกจากโปรแกรมไปสู่ Dos แต่ถ้ากด N ก็จะเข้าสู่เมนูหลักอีกครั้ง แสดงหน้าจอการออกจากโปรแกรมได้ดังรูปที่ 5.11



รูปที่ 5.9 แสดงภาพหน้าจอการควบคุมมอเตอร์



รูปที่ 5.10 แสดงภาพหน้าจอส่วนของ Information

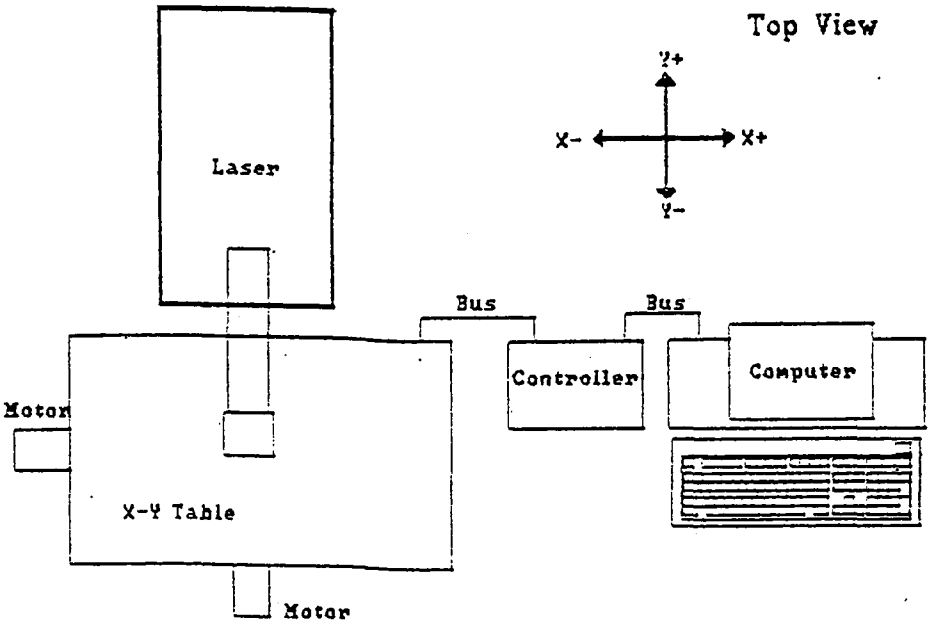


รูปที่ 5.11 แสดงภาพหน้าจอการออกจากโปรแกรม

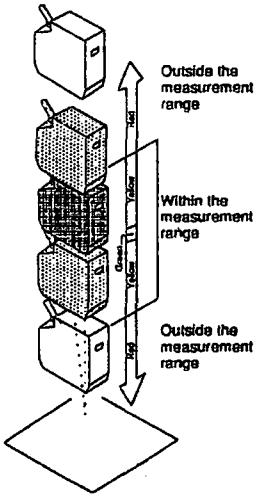
บทที่ 6

ผลการทดลอง

6.1 ขั้นตอนการทดลอง



รูปที่ 6.1 แสดงการวางตำแหน่งอุปกรณ์ต่าง ๆ และแสดงเทียบกับทิศการตั้งแกน



รูปที่ 6.2 แสดงระยะการทำงานของ Laser Displacement Sensor

1. จัดวางอุปกรณ์ดังรูปที่ 6.1โดยทำการติดตั้ง Laser Displacement Sensor

เข้ากับตัวจับ โดยต้องปรับระยะสูงจากพื้นผิวตามความเหมาะสมของลักษณะและขนาดพื้นผิว โดย Laser Displacement Sensor จะทำงานอยู่ในช่วงห่างจากวัตถุ 6 ซม. แต่ไม่เกิน 14 ซม. จากวัตถุโดยแสดงช่วงระยะการทำงานของ LDS ดังรูปที่ 6.2

2. นำวัตถุที่จะทำการสแกนพื้นผิววางบน XY-Table พร้อมทั้งจัดตำแหน่งในการเริ่มสแกนให้เหมาะสม

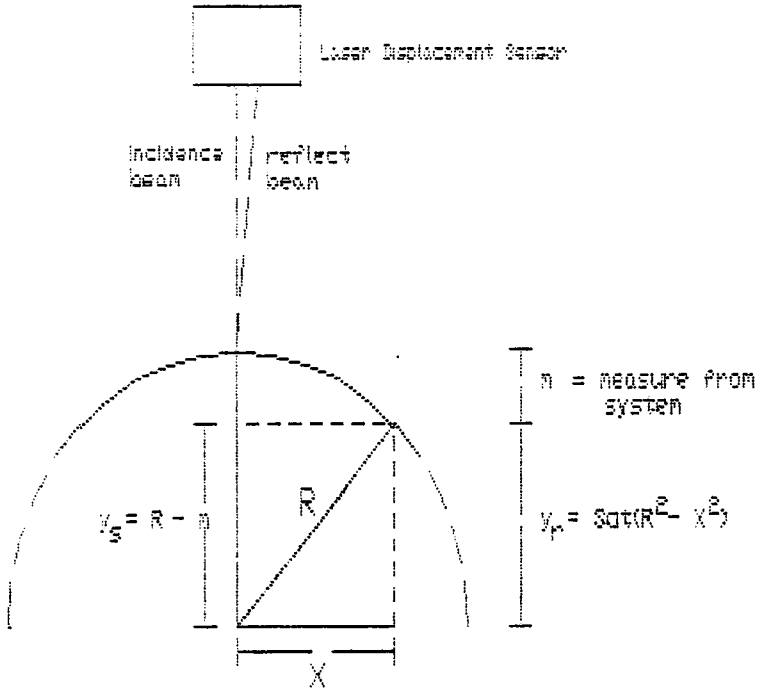
3. เริ่มทำการสแกน ในโครงงานพิเศษนี้สามารถเลือกค่าความละเอียดในการเก็บข้อมูลได้ (เป็นการเลื่อนของ XY-Table ต่อการเก็บข้อมูล 1 จุด) โดยมีข้อควรคำนึงถึงขนาดของข้อมูลซึ่งสามารถทำการเก็บได้คือ ไม่เกิน 350 * 350 จุด

4. นำการเก็บข้อมูลเข้าคอมพิวเตอร์ และทำการสร้างภาพพื้นผิวของวัตถุเพื่อนำมาวิเคราะห์

6.2 ผลการทดลองและวิจารณ์

6.2.1 การหาค่า Resolution ของการวัดในแนวตั้ง

เนื่องจากไม่สามารถหาตัวอย่างพื้นผิวที่มีระดับสูงต่ำน้อย ๆ ในระดับ 10 μm. ซึ่งเป็นค่า Resolution ของ LDS จึงได้ออกแบบการทดลองประกอบการคำนวณโดยทำการวัดส่วนของพื้นผิวทรงกลมซึ่งทราบรัศมีที่แน่นอน อธิบายการทดลองดังรูปที่ 6.3



รูปที่ 6.3 แสดงการคำนวณหาค่า Resolution ของการวัดในแนวตั้ง

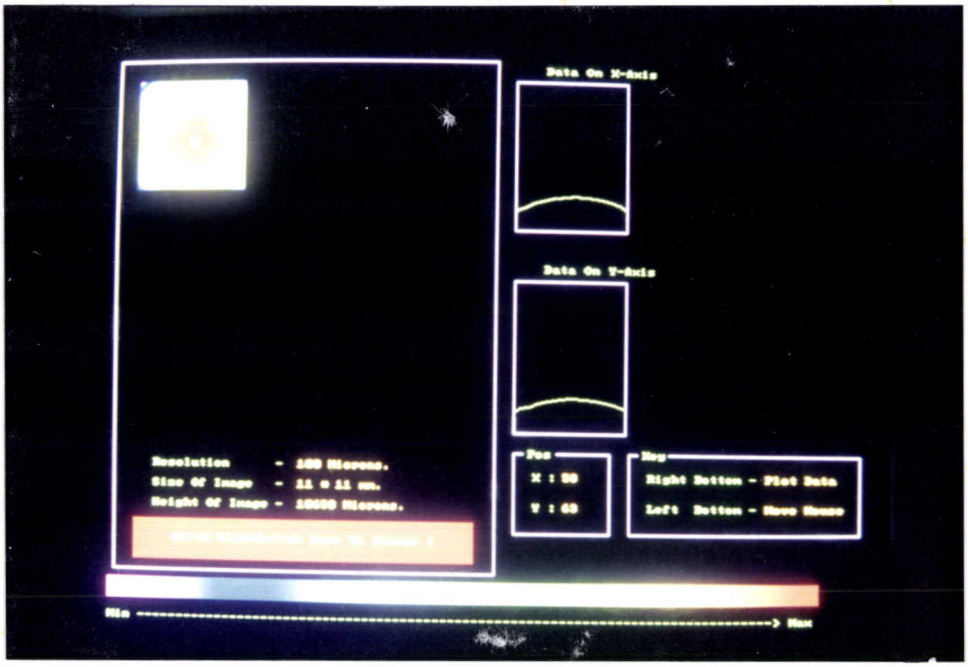
จากรูป สามารถหาความสูง y ซึ่งถือว่าเป็นความสูงจริงในแนวตั้งจะสัมพันธ์กับระยะทางการสแกนแนวแกน x กับรัศมีของทรงกลม จะได้ระยะสูงจริง

$$y_r = Sqr(R^2 - X^2)$$

เมื่อพิจารณาค่าความสูงของวัตถุ m ที่ได้จากการสแกนพื้นผิวจะได้ความสัมพันธ์กับรัศมี R คือ ระยะสูงเนื่องจากการวัด

$$y_m = R - m$$

นำค่าความสูงที่หาได้ทั้งสองแบบนำมาเปรียบเทียบกัน ประกอบการพิจารณาจากการวิเคราะห์ภาคตัดขวางของภาพที่ได้จากการสแกน

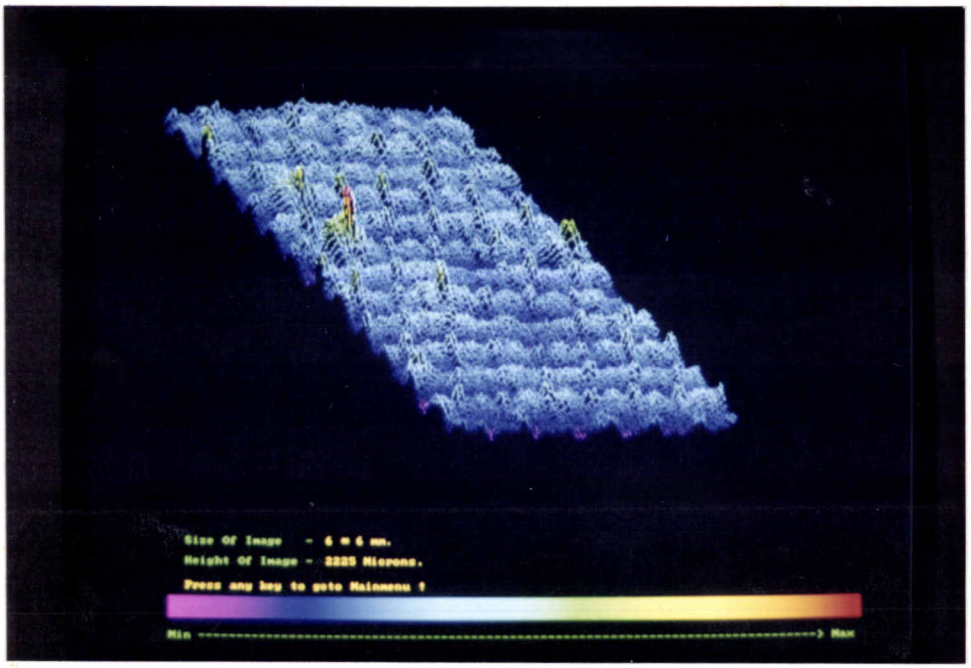


รูปที่ 6.4 แสดงภาคตัดขวางของส่วนของพื้นผิวทรงกลมที่นำมาคำนวณ

ในรูปที่ 6.4 จะเห็นว่า ข้อมูลที่ได้เมื่อนำมาสร้างภาพจะได้ภาพส่วนของความโค้งได้ชัดเจน จึงอนุมานได้ว่าความสูง m ที่ได้มีความถูกต้องพอที่จะนำมาคำนวณจากวิธีที่กล่าวมาข้างต้น ผลที่ได้ค่าความสูงที่ได้จากการวัดจะค่าความผิดพลาดจากระยะความสูงจริงประมาณ 3 % ดังนั้นจะได้ค่า Resolution ใกล้กับ 10 μm. คือประมาณ 10.3 μm.

6.3 การหาความละเอียดของการวัดในแนวนอน

ในโครงงานพิเศษนี้ได้ทดลองนำตัวอย่างที่มีขนาดเล็กและมีลักษณะเด่นชัดมาสแกนที่ระยะความละเอียด 10 ไมครอน สามารถแสดงผลได้ค่อนข้างถูกต้องและชัดเจน



(a)

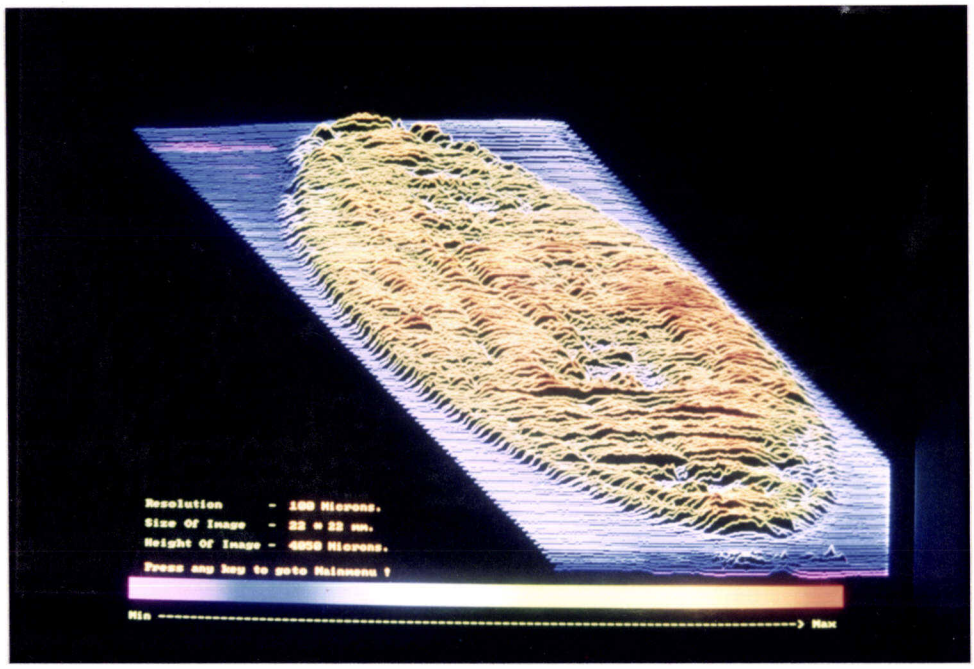
(b)

รูปที่ 6.5 แสดงภาพที่ได้จากการสแกนตาข่ายลวดขนาดเล็ก

(a) แสดงแบบวิเคราะห์ภาคตัดขวาง

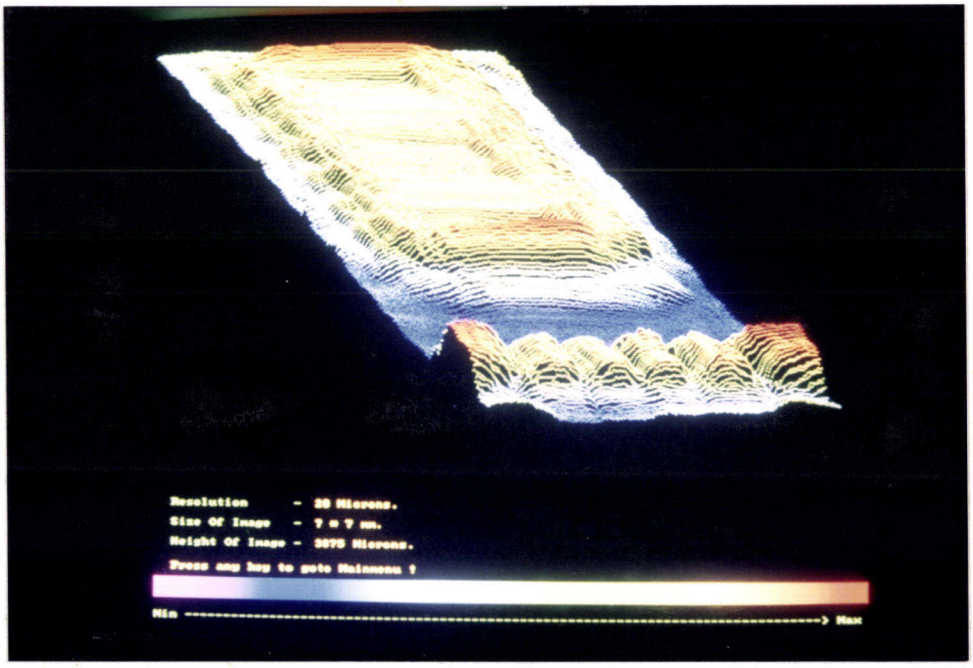
(b) แสดงแบบ 3 มิติ

6.4 ตัวอย่างภาพพื้นผิววัตถุที่ได้ทำการตรวจสอบ



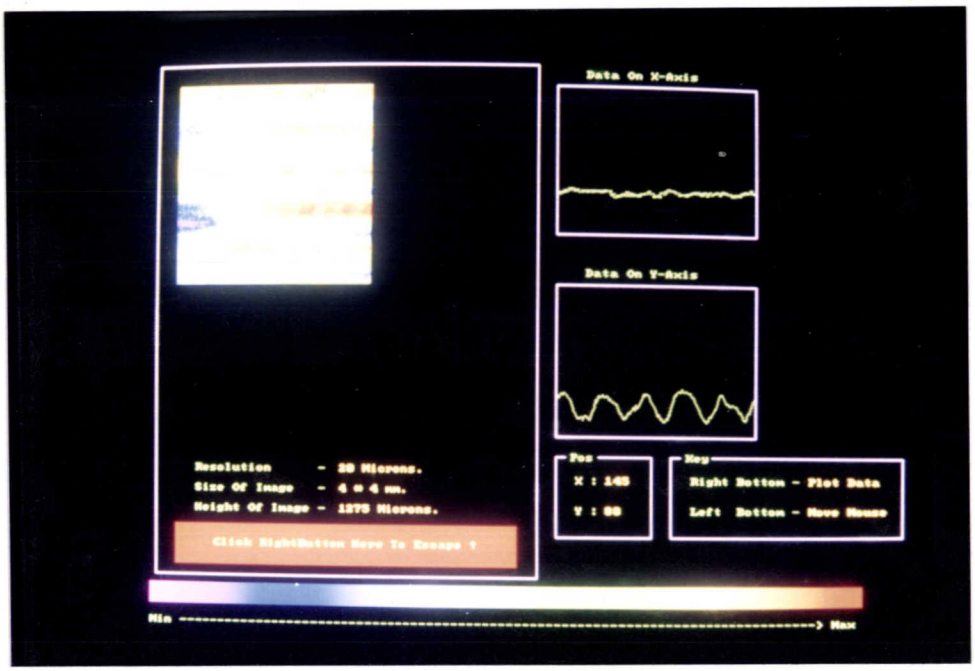
รูปที่ 6.6 ด้านก้อยของเหรียญบาท

- ขนาด 23*23 มม.
- ความละเอียด 100 ไมครอน



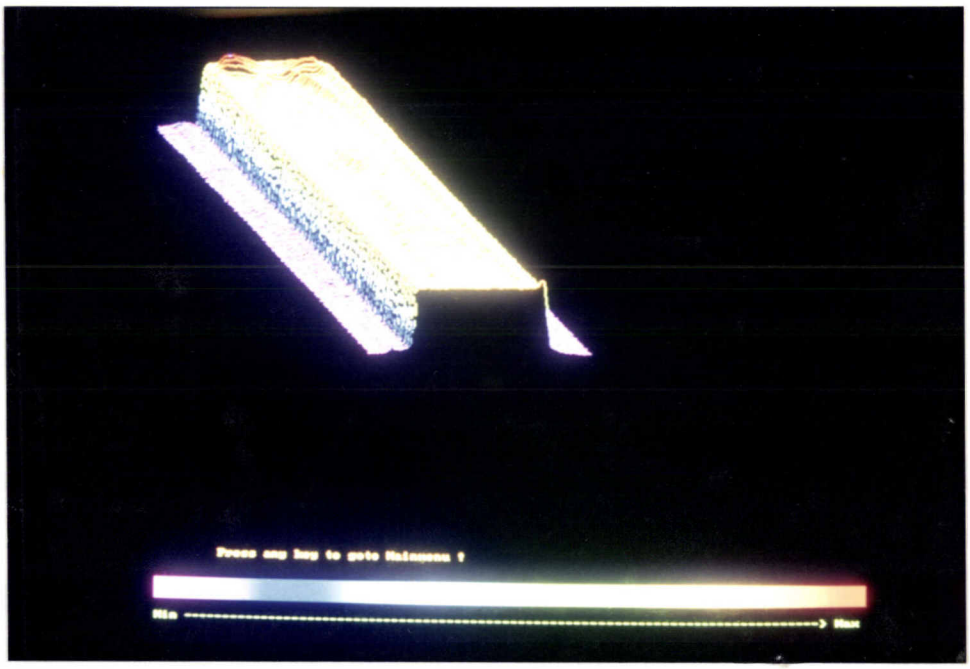
รูปที่ 6.7 ไมโครชิปของ Eprom

- ขนาด 7*7 มม.
- ความละเอียด 20 ไมครอน



รูปที่ 6.8 เส้นแสดงสเกล 1/64 นิ้วของไม้บรรทัดเหล็ก

- ขนาด 4*4 มม.
- ความละเอียด 20 ไมครอน



รูปที่ 6.9 พระเครื่อง (สมเด็จวัดระฆัง)

- ขนาด 23*23 มม.
- ความละเอียด 100 ไมครอน

บทที่ 7

สรุปผลและวิจารณ์

สรุปผล

1. จากการนำข้อมูลที่ได้จากการสแกนพื้นผิว มาทำการแสดงผลโดยโปรแกรมการแสดงภาพที่ได้พัฒนาขึ้น สามารถแสดงผลได้ถูกต้อง ใกล้เคียงลักษณะของพื้นผิววัตถุที่นำมาสแกน

2. ความละเอียดในแนวแกนตั้งของระบบตรวจสอบพื้นผิววัตถุด้วยเลเซอร์ สามารถจำแนกความแตกต่างของความขรุขระในระดับ 10 ไมครอนได้

3. ความละเอียดในแนวแกนนอนของระบบตรวจสอบพื้นผิววัตถุด้วยเลเซอร์สามารถที่จะจำแนกลักษณะของพื้นผิวตัวอย่างในระดับ 10 ไมครอนได้ชัดเจนพอสมควร

4. วัตถุที่พื้นผิวสะท้อนแสงและมีความชันของระดับพื้นผิวมาก ๆ จะส่งผลให้การหาความสูงรวมของภาพและการแสดงภาพเกิดความผิดพลาดได้

วิจารณ์

1. การสแกนพื้นผิววัตถุที่ต้องการความละเอียดในแนวนอนมาก ๆ ทำให้ข้อมูลที่เก็บมีขนาดใหญ่ ไม่สามารถที่จะนำไปแสดงผลได้ มีผลให้เก็บข้อมูลจากขนาดพื้นที่ของพื้นผิววัตถุตัวอย่างน้อยลง

2. ความละเอียดในแนวนอน จะขึ้นกับขนาดเส้นผ่านศูนย์กลางของลำเลเซอร์ ที่ตกกระทบลงบนพื้นผิวตัวอย่าง ซึ่งใน Laser Displacement Sensor ที่ใช้ในโครงงานพิเศษนี้ จะมีขนาด 1.0 * 2.0 มม^2.

3. ค่าความผิดพลาดที่พื้นผิววัตถุที่สะท้อน เกิดจากมีการสะท้อนแสงและกระเจิงออกไปตกที่ PSD ได้น้อย ทำให้ตัว Controller ของเลเซอร์ทำงานในโหมดชดเชย Gain ขยายของค่าแรงดันเอาท์พุทผิดพลาด แต่ถ้าพื้นผิววัตถุมีค่าการสะท้อนเท่ากันหมด จะทำค่าผิดพลาดลดน้อยลง ส่วนในกรณีค่าผิดพลาดจากพื้นผิววัตถุมีความชันมาก ๆ หรือเป็นขอบตัดมีสาเหตุเกิดจาก แสงเลเซอร์ไม่สะท้อนกลับลงบน PSD ทำการแก้ไขได้โดยทำการสแกนพื้นผิวที่อยู่ในระนาบเดียวเท่านั้น

ข้อเสนอแนะ

การตรวจสอบพื้นผิววัตถุโดยใช้แสงเลเซอร์นั้น ความละเอียดของการวัดจะขึ้นกับคุณสมบัติของ Laser Displacement Sensor (LDS) เป็นสำคัญ คือจะต้องมีขนาดของเส้นผ่านศูนย์กลางของลำเลเซอร์เล็ก ๆและมีความไวของ PSD(Position Sensitive Device) สูง ส่วนในระบบขับเคลื่อนนั้นต้องมีความแม่นยำต่อตำแหน่งสูงตามการวัดที่ต้องการความละเอียดเพิ่มขึ้น ในส่วนการเก็บข้อมูลและแสดงผลนั้นควรมีการจัดการกับข้อมูลที่มีขนาดใหญ่โดยบีบอัดข้อมูลให้มีขนาดเล็กลง เพื่อเพิ่มความละเอียดในการเก็บข้อมูลได้

ภาคผนวก ก.

โปรแกรม

```
Program Project;
Uses Crt,Dos,Graph,Sim;
Type     Menu = Array[1..5] Of String;
     String50 = String[50];
      Ar_Data = Array[1..16] Of Byte;
Const  MaxMenu = 5;
       Item : Menu = ('Scan',
                      'Get Image',
                      'Control Motor',
                      'Information',
                      'Quit');

       PA  = $0300;  PA1  = $0304;
       PB  = $0301;  PB1  = $0305;
       PC  = $0302;  PC1  = $0306;
       PCC = $0303; PCC1  = $0307;

       Data_Out : Ar_Data = ($09,$03,$06,$0C,$0C,$06,$03,$09,
                             $90,$30,$60,$C0,$C0,$60,$30,$90);

Var d,dir,i,j,k,n,o,p,q,r,s,x,y,z,OldY,OldI,ForG,BacG : Integer;
    CwIndex,CcwIndex,YcwIndex,YccwIndex,Mode,T : Byte;
    Hr,Min,Sec,Sec100,Color,Size : Word;
    Mono,VgaMono,TisMode,Check   : Boolean;
    Index,Ind,A,B : Byte; Pt : Pointer;
    Ch : Char; Msg  : String50;
    CurX,CurY,NoStr,Code : Integer;
    Sentence : String[8]; Num : String[4];

            {********** Title ***********}
```

```
Procedure Opengraph;
Var GrDriver,GrMode : Integer;
Begin
  Mono := False; VgaMono := False;
  Color := 3;
  DetectGraph(GrDriver,GrMode);
   Case GrDriver Of
     5 : VgaMono := True;
     7 : Begin Mono := True; Color := 1; End;
     9 : GrMode := 2;
   End;
  InitGraph(GrDriver,GrMode,'');
End;

Procedure LoadFont(FontName :String; Size:Byte);
Begin
  T := InstallUserFont(FontName);
  SetTextStyle(T,0,Size);
End;

Procedure OutTextSha(x,y : Integer; St : String);
Begin
  SetColor(0);  OutTextXY(x+3,y+4,St);
  SetColor(15); OutTextXY(x,y,St);
End;

Procedure OutTxt(x,y:Integer; Size:Byte; Color:Word; Msg:String);
Begin
  SetColor(Color); SetTextStyle(0,0,Size);
  OutTextXY(x,y,Msg);
End;
```

```
Procedure LoadPic(x,y : Integer);
Var Size : Word; P,T : Pointer;
      FP : File;
Begin
  Assign(FP,'Kmitl.vga');
  {$I-}
  Reset(FP,1);
  {$I+};
  If IoResult <> 0 Then
   Begin
     Closegraph;
     Writeln(' File KMITL.VGA not found !');
     Halt(0);
   End;
  Bar(x-1,y-1,x+177,y+171);
  Size := FileSize(FP);
  GetMem(P,Size);
  BlockRead(FP,P^,Size);
  PutImage(x,y,P^,NormalPut);
  Close(FP);
  FreeMem(P,Size);
End;

Procedure Date;
Var Yr,Mo,Day,DayOfWk : Word;
    YrStr,MoStr,DayStr : String;
    Indic : Integer;
Begin
  GetDate(Yr,Mo,Day,DayOfWk); Indic := DayOfWk;
  Str(Yr,YrStr);Str(Mo,MoStr);Str(Day,DayStr);
   Case Indic Of
```

```
    0:OutTextXY(510,67,'Sunday');    1:OutTextXY(510,67,'Monday');
    2:OutTextXY(510,67,'Tuesday');   3:OutTextXY(510,67,'Wednesday');
    4:OutTextXY(510,67,'Thursday');  5:OutTextXY(510,67,'Friday');
    6:OutTextXY(510,67,'Saturday');
   End;
  OutTextXY(505,77,DayStr+'-'+MoStr+'-'+YrStr);
End;

Procedure Frame0(x1,y1,x2,y2 : Integer; BackGr : Word);
Begin
  SetFillStyle(1,BackGr);  Bar(x1,y1,x2,y2);
  SetColor(15); Rectangle(x1,y1,x2,y2);
End;

Procedure Frame1(x1,y1,x2,y2 : Integer; BackGr : Word);
Begin
  SetFillStyle(1,BackGr); Bar(x1,y1,x2-10,y2-10);
  SetColor(0); Rectangle(x1+5,y1+5,x2-15,y2-15);
  SetFillStyle(1,0);
  Bar(x2-10,y1+10,x2,y2); Bar(x1+10,y2-10,x2,y2);
  SetColor(15); Rectangle(x1,y1,x2-10,y2-10);
End;

Procedure Frame2(x1,y1,x2,y2 : Integer; BackGr : Word);
Begin
  SetFillStyle(1,BackGr); Bar(x1,y1,x2-10,y2-10);
  SetFillStyle(1,0);
  Bar(x2-10,y1+10,x2,y2); Bar(x1+10,y2-10,x2,y2);
  SetColor(15); Rectangle(x1,y1,x2-10,y2-10);
End;
```

```
Procedure Frame3(x1,y1,x2,y2 : Integer; BackGr : Word);
Begin
  SetFillStyle(1,BackGr);
  Bar(x1+1,y1+1,x2-1,y2-1);
End;

Procedure SubTit(x1,y1,x2,y2,Space : Integer; Msg : String);
Begin
  SetColor(15); Rectangle(x1,y1,x2,y2);
  SetFillStyle(1,9); Bar(x1+10,y1-12,x1+Space,y1+5);
  OutTxt(x1+15,y1-3,1,14,Msg); SetColor(15);
End;

Procedure Key;
Begin
  SubTit(260,310,590,380,40,'Key');
  OutTxt(275,330,1,14,'[Esc]          [Enter] ');
  OutTxt(275,330,1,15,'      - Quit          - Continue');
  OutTxt(275,350,1,14,'[Up] & [Down] ');
  OutTxt(275,350,1,15,'              Arrow - Select other menu');
End;

Procedure Title;
Begin
  Frame0(0,0,GetMaxX,GetMaxY,5); LoadPic(30,40);
  LoadFont('Trip',5); OutTextSha(342,50,'LASER');
  LoadFont('Trip',5); OutTextSha(225,105,'SURFACE SCANNING');
  LoadFont('Trip',5); OutTextSha(330,155,'SYSTEM');
  LoadFont('Trip',1); OutTextSha(120,220,'By..');
  OutTextSha(180,250,'Mr.Rungrueng Saekow  33504030');
  OutTextSha(180,280,'Mr.Somsak Puntuwong 33504038');
```

```
  OutTextSha(188,310,'Department Of Applied Physics ');
  OutTextSha(230,340,'Faculty Of Science');
  OutTextSha(90,370,'King Mongkut'#39's Institute Of Technology Ladkrabang');
  Ch := Readkey; Delay(500);
End;

Procedure Title1;
Begin
  Frame0(0,0,GetMaxX,GetMaxY,1); Frame0(0,0,GetMaxX,25,13);
  OutTxt(200,10,1,15,'Laser Surface Scanning System ');
  Frame0(35,90,210,380,9);  Frame0(250,90,600,420,9);
  Frame0(35,90,210,120,13); OutTextXY(108,103,'Menu');
  Frame0(250,60,600,90,13); Frame0(480,60,600,90,13);
  OutTxt(325,70,1,15,'Describtion');
End;

            {********* End of Title ********}

Procedure SaveScreen(x1,y1,x2,y2 : Integer);
Begin
  Size := ImageSize(x1,y1,x2,y2);
  GetMem(Pt,Size);
  GetImage(x1,y1,x2,y2,Pt^);
End;

Procedure RestoreScreen(x1,y1 : Integer);
Begin
  PutImage(x1,y1,Pt^,0);
End;

Procedure Describe1;
```

```
Begin
  Frame3(250,90,600,270,9);
  SubTit(260,110,590,255,50,'Scan');
  OutTextXY(285,130,'Before scan you must input Filename,');
  OutTextXY(275,150,'Range and Resolution.');
  OutTxt(285,170,1,14,'Filename ');
  OutTxt(285,170,1,15,'         - 8 Characters Maximum.');
  OutTxt(285,190,1,14,'Range ');
  OutTxt(285,190,1,15,'      - Not more than 100 Milimetres.');
  OutTxt(285,210,1,14,'Resolution ');
  OutTxt(285,210,1,15,'           - 10,20,40,100,200,250,500');
  OutTxt(285,230,1,15,'             or 1000 Microns Only.');
End;

Procedure Describe2;
Begin
  Frame3(250,90,600,260,9);
  SubTit(260,110,590,180,130,'Get Image File');
  OutTextXY(285,130,'To Display 3-D Image by load image ');
  OutTextXY(275,150,'Filename.');
End;

Procedure Describe3;
Begin
  Frame3(251,91,599,270,9);
  SubTit(260,110,590,200,195,'Control Stepping Motor');
  OutTextXY(285,130,'You can control stepping motor to set ');
  OutTextXY(275,150,'position of subject which you want to');
  OutTextXY(275,170,'scan by using arrow key.');
End;
```

```
Procedure Describe4;
Begin
  Frame3(250,90,600,270,9);
  SubTit(260,110,590,200,110,'Information');
  OutTextXY(285,130,'This is the information about this ');
  OutTextXY(275,150,'project. Press [Enter] to get more ');
  OutTextXY(275,170,'information.');
End;

Procedure Describe5;
Begin
  Frame3(251,91,599,270,9);
  SubTit(260,110,590,180,50,'Quit');
  OutTextXY(275,130,'If you want to return to Dos then');
  OutTextXY(275,150,'press [Enter] to continue.');
End;

Procedure CtlText(x,y : Integer; Forg,Bacg : Byte; Msg : String);
Var PosX : Integer;
Begin
  SetFillStyle(1,Bacg); SetColor(Forg);
  OutTextXY(x,y,Msg);
End;

Function PopUp(x,y,Forg,Bacg : Integer; Item : Menu): Byte;

Procedure ChangeMenu;
Begin
  SetFillStyle(1,Bacg); Bar(x-10,y-5,190,y+10);
  SetFillStyle(1,9);    Bar(x-10,OldY-5,190,OldY+10);
  CtlText(x,OldY,Forg,Bacg,Item[OldI]);
```

```
  CtlText(x,y,ForG,BacG,Item[Index]);
  SetFillStyle(1,Bacg);
End;

Begin
  Repeat
   Ch := Readkey;
     If Ch = #0 Then
      Begin
        Ch := Readkey;
        Case Ch Of.
          #72 : Begin
                  OldI := Index; Index := Index - 1;
                  If Index = 0 Then Index := 5;
                  y := 150 + ((Index -1) * 50);
                  OldY := y + 50;
                  If y = 350 Then OldY := 150;
                  ChangeMenu;
                End;
          #80 : Begin
                  y := 150 + (Index * 50); OldY := y -50;
                  OldI := Index; Index := Index + 1;
                  If OldY = 350 Then y := 150;
                  If Index = 6 Then Index := 1; ChangeMenu;
                End
          Else If (Index <> OldI) Then  z := 1;
        End;
      End Else z := 1;
     If (Index <> OldI) And (z = 0) Then
       Case Index Of
         1 : Describe1;  2 : Describe2;
```

```
        3 : Describe3;  4 : Describe4;
        Else Describe5;
      End;
      z := 0;
  Until (Ch = #13) Or (Ch = #27);
    If (Index = 1) And (Ch = #13) Then PopUp := 1 Else
    If (Index = 2) And (Ch = #13) Then PopUp := 2 Else
    If (Index = 3) And (Ch = #13) Then PopUp := 3 Else
    If (Index = 4) And (Ch = #13) Then PopUp := 4 Else
    If (Ch = #27) Or ((Index = 5) And (Ch = #13)) Then
       PopUp := 0;
End;

Procedure CursorOn;
Begin
   SetFillStyle(1,0); Bar(CurX,CurY+5,CurX+5,CurY+4);
End;

Procedure CursorOff;
Begin
  SetFillStyle(1,15); Bar(CurX,CurY+5,CurX+5,CurY+4);
End;

Procedure InputName;
Begin
  SetColor(0); CursorOn; NoStr := 0; Sentence := '';
  Repeat
   Ch := Readkey;
   If (Ch = #8) And (NoStr <> 0) Then
    Begin
      CursorOff; CurX := CurX - TextWidth(Ch);
```

```
      Bar(CurX,CurY+5,CurX+7,CurY-3); CursorOn;
      Delete(Sentence,NoStr,1);  NoStr := NoStr - 1;
    End;
    If (NoStr < 8) And (Ch <> #8) And (Ch <> #13) And
       (Ch <> #0) And (Ch <> #27) Then
       Begin
        CursorOff; OutTextXY(CurX,CurY-2,Ch);
        CurX := CurX + TextWidth(Ch); CursorOn;
        Sentence := Sentence + Ch;  NoStr := NoStr + 1;
       End;
  Until Ch = #13;
  CursorOff; p := 1;
End;

Procedure InputNum;
Begin
  SetColor(0); CursorOn; NoStr := 0; Num := '';
  Repeat
   Ch := Readkey;
   If (Ch = #8) And (NoStr <> 0) Then
    Begin
     CursorOff; CurX := CurX - TextWidth(Ch);
     Bar(CurX,CurY+5,CurX+7,CurY-3);
     CursorOn; Delete(Num,NoStr,1);
     NoStr := NoStr - 1;
    End;
   If (NoStr < 4) And (Ch in ['0'..'9']) Then
    Begin
     CursorOff; OutTextXY(CurX,CurY-2,Ch);
     CurX := CurX + TextWidth(Ch);
     CursorOn; Num := Num + Ch;
```

```
      NoStr := NoStr + 1;
    End;
  Until Ch = #13;
  CursorOff;
  CurX := x; CurY := y; p := 1;
End;

          {************ Scan ************}

Procedure Scan;
Var x,y,OldY,Index,OldIndex,Processed : Integer;
    Range,cc,nn,dd,r,s,d,i,j,b,g,w : Integer;
    Data1,Data2,Data3 : String[8];
    Filename : String[12]; Step : Real;
    Hr1,Min1,Sec1,Sec1001,Hr2,Min2,Sec2,Sec1002 : Word;
    Size,Resol : Word;
    HrStr,MinStr,SecStr : String; Ptr : Pointer;
    Fi : File Of Word;
    StrB,StrE,StrX,StrY,Path,PathFile : String;
    Data : Array[0..4000] Of Word;
    Work,Cancel : Boolean;
    DirInfo : SearchRec;

Procedure WriteData;
Begin
  If (p <> 0) Then
   Begin
     SetColor(3);
     If Data1 <> '' Then
       Begin
         OutTextXY(390,130,Filename);
```

```
        End;
      If Data2 <> '' Then
        Begin
          OutTextXY(390,160,Data2);
        End;
      If Data3 <> '' Then
        Begin
          OutTextXY(390,190,Data3);
        End;
    End;
End;

Procedure StMotor(P,Q:Byte);
Var  N,J : Integer;
Begin
  For N := 1 To NN Do
    For J := P To Q Do
     Begin
       Port [PA1]:= Data_Out[J];
       Delay (7);
     End
End;

Procedure AtoD;
Var A,C,E :Integer;
    Num,Status : Byte;
    Sum : Word;
Begin
  Delay(550);
  Sum := 0;
  For Num := 1 To 15 Do
```

```
    Begin
     Port [$308] := $FF;
     Port [PCC1] := $83 ;
     Repeat
       Status := Port [PB1];
       Status := Status And $01;
     Until Status = $00;
     A := Port [$308];
     C := Port [$309];
     E := A Shl 4 + C Shr 4;
     Sum := Sum + E;
    End;
    Sum := Sum Div 15; Str(Sum,StrE);
    If Odd(I) Then B := B+1
    Else B := B-1;
    cc := cc+1;
    Data[cc] := Sum; Str(B,StrB);
    Write(Fi,Data[cc]);
    SetFillStyle(1,9);
    Bar(310,330,350,350); Bar(485,330,530,350);
    OutTxt(315,340,1,15,StrB); OutTxt(490,340,1,15,StrE);
End;

Begin
  SaveScreen(251,91,599,419);
  Frame3(251,91,599,419,9);
  SubTit(260,110,590,270,50,'Scan');
  OutTxt(275,130,1,14,'Filename   :');
  OutTextXY(275,160,'Range      :');
  OutTextXY(275,190,'Resolution :');
  SubTit(260,310,590,380,40,'Key');
```

```
OutTxt(275,330,1,14,'[Esc]        [Spacebar]        ');
OutTxt(275,330,1,15,'      - Quit            - Select');
OutTxt(275,350,1,14,'[Up] & [Down] ');;
OutTxt(275,350,1,15,'              Arrow - Select choice');
x := 390; y := 130; z := 0; BacG := 15; Index := 1;
Work := False; Cancel := False;:
SetFillStyle(1,Bacg); Bar(x-10,y-5,490,y+7);
Repeat
 Ch := Readkey;
 If Ch = #0 Then
  Begin
   Ch := Readkey;
    Case Ch Of
    #72 : Begin
           OldY := y;  y := y - 30;
           Index := Index - 1;
           If Index = 0  Then Index := 4;
           If Index = 4  Then y := 230;
           If OldY = 230 Then y := 190;
           SetFillStyle(1,15); Bar(x-10,y-5,490,y+7);
           SetFillStyle(1,9);  Bar(x-10,OldY-5,490,OldY+7);
           SetFillStyle(1,15);
           If Index = 4 Then
            Begin
             OutTxt(380,230,1,4,' READY (Y/N)');
             SetColor(3);
            End;
            WriteData;
          End;
    #80 : Begin
           OldY := y; y := y + 30;
```

```
          Index := Index + 1;
          If Index = 4 Then y := 230;
          If Index = 5 Then Index := 1;
          If OldY = 230 Then y := 130;
           SetFillStyle(1,15); Bar(x-10,y-5,490,y+7);
           SetFillStyle(1,9);  Bar(x-10,OldY-5,490,OldY+7);
           SetFillStyle(1,15);
           If Index = 4 Then
            Begin
              OutTxt(385,230,1,4,' READY (Y/N)');
              SetColor(3);
            End;
            WriteData;
          End;
   End;
 End;
If Ch = #32 Then
  Case Index Of
  1 : Begin
       SetFillStyle(1,Bacg); Bar(x-10,y-5,490,y+7);
       CurX := 390; CurY := 130;
       InputName; Data1 := Sentence;
       Filename := Data1 + '.img';
      End;
  2 : Begin
       Repeat
        SetFillStyle(1,Bacg); Bar(x-10,y-5,490,y+7);
        CurX := 390; CurY := 160; InputNum;
        Data2 := Num; Val(Data2,Range,Code);
       Until ((Range > 0) And (Range < 101));
       Data2 := Data2 + ' mm.';
```

```
      End;
  3 : Begin
      Repeat
       SetFillStyle(1,Bacg); Bar(x-10,y-5,490,y+7);
       CurX := 390; CurY := 190; InputNum;
       Data3 := Num; Val(Data3,Resol,Code);
      Until (Resol = 10)  Or (Resol = 20)  Or (Resol = 40)  Or
            (Resol = 100) Or (Resol = 200) Or (Resol = 250) Or
            (Resol = 500) Or (Resol = 1000);
      Data3 := Data3 + ' Mic';
     End;
  4 : Begin
      SetFillStyle(1,15); Bar(x-10,y-5,490,y+7);
      OutTxt(380,230,1,4,'  Y     N  ');
      OutTxt(380,230,1,0,'   es Or  o ');
       Repeat
        Ch := Readkey;
        If (UpCase(Ch) = 'Y') And (Data1 <> '') And
           (Data2 <> '')  And (Data3 <> '') Then
         Begin
           Frame3(251,91,599,419,9);
           SubTit(260,110,590,240,50,'Scan');
           OutTxt(275,130,1,14,'Filename   :');
           OutTextXY(275,150,'Range      :');
           OutTextXY(275,170,'Resolution :');
           OutTextXY(275,190,'Start Time :');
           OutTextXY(275,210,'End Time   :');
           GetTime(Hr1,Min1,Sec1,Sec1001);
           Str(Hr1,HrStr); Str(Min1,MinStr); Str(Sec1,SecStr);
           OutTxt(390,130,1,15,Filename);
           OutTextXY(390,150,Data2); OutTextXY(390,170,Data3);
```

```
OutTextXY(390,190,HrStr+':'+MinStr+':'+SecStr);
SubTit(260,265,590,395,100,'Processing');
OutTextXY(275,285,'0%          50%          100%');
SetFillStyle(1,1); Bar(275,300,575,320);
OutTxt(355,340,1,14,'Y : ');
OutTxt(385,340,1,15,'0');
OutTxt(285,340,1,14,'X : ');
OutTxt(430,340,1,14,'Data :');
B:=1; Y:=0; R:=9; S:=12;
Case Resol Of
   10 : Begin D := 100 * Range; NN :=   1; End;
   20 : Begin D :=  50 * Range; NN :=   2; End;
   40 : Begin D :=  25 * Range; NN :=   4; End;
  100 : Begin D :=  10 * Range; NN :=  10; End;
  200 : Begin D :=   5 * Range; NN :=  20; End;
  250 : Begin D :=   4 * Range; NN :=  25; End;
  500 : Begin D :=   2 * Range; NN :=  50; End;
 1000 : Begin D :=   1 * Range; NN := 100; End;
End;
Assign(Fi,Filename);
Rewrite(Fi);
Data[0]:=D+1;
Write(Fi,Data[0]);
For I := 1 To D Do
 Begin
   Step := I * (300/D); Processed := Trunc(Step);
   SetFillStyle(1,11);
   Bar(275,300,275+Processed,320);
   CC:=0;
   If Odd(I) Then
    Begin
```

```
  G:=1; W:=4; B:=B-1;
End
Else
Begin
  G:=5; W:=8; B:=B+1;
End;
AtoD;
For J := 1 To D Do
 Begin
  If KeyPressed Then
   Begin
     Size := ImageSize(220,190,440,280);
     GetMem(Ptr, Size);
     GetImage(220,190,440,280,Ptr^);
     Frame1(220,190,440,280,4);
     OutTextXY(250,210,'Interrupt By User');
     OutTextXY(240,240,'Cancel Process ? (Y/N)');
     Repeat
      Ch := Readkey;
      If UpCase(Ch) = 'Y' Then
       Begin
         PutImage(220,190,Ptr^,0);
         FreeMem(Ptr,ImageSize(220,190,440,280));
         I := D; Cancel := True;
       End;
      If UpCase(Ch) = 'N' Then
       Begin
         Work := False;
         PutImage(220,190,Ptr^,0);
         FreeMem(Ptr,ImageSize(220,190,440,280));
       End;
```

```
        Until (UpCase(Ch) = 'Y') Or (UpCase(Ch) = 'N');
      End;
     If Cancel = False Then
      Begin
        Port[PCC1] := $80;
        StMotor(G,W);
        AtoD;
      End;
    End;
 If Cancel = False Then
  Begin
   StMotor(R,S);
   Y:=Y+1;
   Str(Y,StrY);
   SetFillStyle(1,9); Bar(380,330,420,350);
   OutTextXY(385,340,StrY);
  End;
 End;
If Cancel = False Then
 Begin
  R:=13;S:=16;
  GetTime (Hr2,Min2,Sec2,Sec1002);
  Str(Hr2,HrStr); Str(Min2,MinStr); Str(Sec2,SecStr);
  OutTxt(390,210,1,15,HrStr+':'+MinStr+':'+SecStr);
  OutTxt(285,365,1,14,'Finished Get Data. Please, Wait !');
  For I := 1 To D Do
   Begin
     StMotor(R,S);
     Y:=Y-1;
   End;
  Bar(270,355,585,375);
```

```
                OutTextXY(275,365,'Completed !  Press Any Key To Mainmenu.');
                Write(Fi,Resol);
                Port [PA1] := $00;
                Close(Fi); Work := True;
                Ch := Readkey; Ch := 'Y'; Delay(500);
               End;
               End;
              If Cancel = True Then
               Begin
                Port [PA1] := $00;
                OutTxt(295,365,1,14,'Cancel Processing ! ');
                Ch := Readkey; Ch := 'Y'; Work := True;
               End;
               If (UpCase(Ch) = 'N') Then
                Begin
                 SetFillStyle(1,15); Bar(x-10,y-5,480,y+7);
                 OutTxt(385,230,1,4,' READY (Y/N)');
                End;
          Until ((UpCase(Ch) = 'Y') And (Data1 <> '') And (Data2 <> '')
                And (Data3 <> '')) Or (UpCase(Ch) = 'N');
          End;
        End; {End Of Case4}
    If Work = True Then Ch := #27;
    If Ch = #27 Then
      Begin
        RestoreScreen(251,91);
        FreeMem(Pt,ImageSize(251,91,599,419));
      End;
  Until Ch = #27;
End;
```

```
          {************ End of Scan ********}


           {*********** Get Image ********}

Procedure Get_Image;
Var Path,PathFile,ImgFile : String;
    x,y,i,j,h,OldY,Index,OldIndex,Created,MaxColor : Integer;
    Data : String[8]; DirInfo : SearchRec;
    Filename : String[12]; YAxis,Level,ImgSize : Integer;
    Fi : File;  Point : Pointer;
    Square : Array[1..4] Of PointType;
    Img : Array[1..361] Of Word; Height : LongInt;
    Data1,Data2,Data3,MethodStr,MouseX,MouseY : String[3];
    Max,Min,MaxDiv,Header,OldMin,Resol : Word;
    x11,y11,x22,y22,Color,YPlot,Method,MouX,MouY,Long : Integer;
    Mouse : MouseType; Mix,Red,Green,Blue : Byte; Divider : Real;
    Pos_MouseX,Pos_MouseY,LongStr,HeightStr,ResolStr : String;

{$F+}
Function DetectSvga : Integer;
Begin
  DetectSvga := 3;
End;
{$F-}

Procedure OpengraphSVGA;
Var Gd,Gm : Integer;
    Driver,ErrCode : Integer;
Begin
  Driver := InstallUserDriver('Svga256',@DetectSvga);
  If GraphResult <> GrOk Then
```

```
    Begin
      Writeln('Error installing TestDriver');
      Halt(1);
    End;
  Gd := Detect; Initgraph(Gd,Gm,'');
  ErrCode := GraphResult;
  If ErrCode <> GrOk Then
    Begin
      Writeln('Error during Init: ', ErrCode);
      Halt(1);
    End;
End;

Procedure WriteData1;
Begin
  SetColor(3);
  If Path <> '' Then
    Begin
      OutTextXY(385,130,Path);
    End;
  If (p <> 0) Then
    Begin
      If Filename  <> '' Then OutTextXY(385,160,Filename);
      If MethodStr <> '' Then OutTextXY(385,190,MethodStr);
    End;
End;

Procedure WriteData2;
Begin
  SetColor(3);
  If (p <> 0) Then
```

```
   Begin
     If Data1 <> '' Then OutTextXY(440,130,Data1);
     If Data2 <> '' Then OutTextXY(440,160,Data2);
     If Data3 <> '' Then OutTextXY(440,190,Data3);
   End;
End;

Procedure Cancel;
Begin
  If KeyPressed Then
   Begin
     Size := ImageSize(270,220,500,310);
     GetMem(Point,Size);
     GetImage(270,220,500,310,Point^);
     If Method = 1 Then
      Begin
        Frame1(270,220,500,310,12);
        SetColor(15);
      End
      Else
      Begin
        Frame1(270,220,500,310,227);
        SetColor(245);
      End;
      OutTextXY(310,240,'Interrupt By User');
      OutTextXY(295,270,'Cancel Process ? (Y/N)');
      Repeat
        Ch := Readkey;
        If UpCase(Ch) = 'Y' Then
         Begin
           PutImage(270,220,Point^,0);
```

```
            FreeMem(Point,ImageSize(270,220,500,310));
            j := h-1; Created := 1;
            If Method = 1 Then
                 OutTxt(40,550,1,14,'Processing Abort By User.')
            Else OutTxt(40,525,1,190,'Processing Abort By User.');
            Delay(500);
          End;
         If UpCase(Ch) = 'N' Then
          Begin
            PutImage(270,220,Point^,0);
            FreeMem(Point,ImageSize(270,220,500,310));
            SetFillStyle(1,0);
            Case Method Of
              1 : SetColor(11);
              2 : SetColor(0);
             End;
           End;
       Until (UpCase(Ch) = 'Y') Or (UpCase(Ch) = 'N');
    End;
End;

Procedure CreatOdd;
Begin
  If Img[i] = 0 Then Img[i] := Min;
  Square[1].x := (i-1)*ImgSize+(j*ImgSize);
  Square[1].y := ((j+YAxis)*ImgSize)-((Img[i-1]-Min) Div MaxDiv);
  Square[2].x := (i-1)*ImgSize+(j*ImgSize);
  Square[2].y := ((j+YAxis)*ImgSize);
  Square[3].x := i*ImgSize+(j*ImgSize);
  Square[3].y := ((j+YAxis)*ImgSize);
  Square[4].x := i*ImgSize+(j*ImgSize);
```

```
  Square[4].y := ((j+YAxis)*ImgSize)-((Img[i]-Min) Div MaxDiv);
  FillPoly(4,Square);
End;

Procedure CreatEven;
Begin
  If Img[i] = 0 Then Img[i] := Min;
  Square[1].x := (h-i)*ImgSize+(j*ImgSize);
  Square[1].y := ((j+YAxis)*ImgSize)-((Img[i-1]-Min) Div MaxDiv);
  Square[2].x := (h-i)*ImgSize+(j*ImgSize);
  Square[2].y := ((j+YAxis)*ImgSize);
  Square[3].x := ((h-1)-i)*ImgSize+(j*ImgSize);
  Square[3].y := ((j+YAxis)*ImgSize);
  Square[4].x := ((h-1)-i)*ImgSize+(j*ImgSize);
  Square[4].y := ((j+YAxis)*ImgSize)-((Img[i]-Min) Div MaxDiv);
  FillPoly(4,Square);
End;

Procedure SetDac (Mix,Red,Green,Blue:Byte);
Begin
  Port[$3c8] := Mix;
  Port[$3c9] := Red;
  Port[$3c9] := Green;
  Port[$3c9] := Blue;
End;

Procedure MixColor;
Var Red,Green,Blue,i,Mix : Byte;
Begin
  Mix := 240;
  For i :=   1 To  48 Do Begin {193-240}
```

```
                        SetDac(Mix,63,Round(i*1.3),0);
                        Dec(Mix);)
                      End;
  For i :=  49 To  96 Do Begin {145-192}
                        SetDac(Mix,Round((96-i)*1.34),63,0);
                        Dec(Mix);
                      End;
  For i :=  97 To 144 Do Begin {97-144}
                        SetDac(Mix,0,63,Round((i-97)*1.34));
                        Dec(Mix);
                      End;
  For i := 145 To 192 Do Begin{49-96}
                        SetDac(Mix,0,Round((192-i)*1.34),63);
                        Dec(Mix);
                      End;
  For i := 193 To 240 Do Begin{1-48}
                        SetDac(Mix,Round((i-193)*1.34),0,63);
                        Dec(Mix);
                      End;
  SetDac(241,0,0,63);   { Blue }
  SetDac(242,0,32,0);   { Green }
  SetDac(243,63,0,0);   { Red }
  SetDac(244,40,40,40); { Gray }
  SetDac(245,63,63,63); { White }
  SetDac(246,32,16,1);  { Brown }
  SetDac(247,32,26,20); { LigthBrown }
  SetDac(248,32,0,32);  { Violet }
  SetDac(249,0,63,0);   { Yellow }
  SetDac(0,0,0,0);      { Black }
End;
```

```
Procedure PlotPixel(x11,y11,x22,y22 : Integer);
Var dx,dy : Integer;
    x,y,m : Real;
Begin
  dy := y11 - y22;
  dx := x11 - x22;
  If Abs(dx) >= Abs(dy) Then
   Begin
     m := dy/dx;
     x := x11; y := y11;
     If x11 <= x22 Then
       While x <= x22 Do
        Begin
          PutPixel(Round(x),Round(y),Color);
          x := x + 1;
          y := y + m;
        End
      Else
       While x >= x22 Do
        Begin
          PutPixel(Round(x),Round(y),Color);
          x := x - 1;
          y := y - m;
        End
   End;
   Else
   Begin
     m := dx/dy;
     x := x11; y := y11;
     If y11 <= y22 Then
       While y <= y22 Do
```

```
        Begin
          PutPixel(Round(x),Round(y),Color);
          y := y + 1;
          x := x + m;
        End
        Else
        While y >= y22 Do
         Begin
           PutPixel(Round(x),Round(y),Color);
           y := y - 1;
           x := x - m;
         End
   End;
End;

Procedure PlotLine;
Begin
  Color := 11;
  PlotPixel(Square[4].x,Square[4].y,Square[1].x,Square[1].y);
End;

Procedure PlotLineColor;
Begin
  Divider := (Max-Min) / 240;
  Color   := (Round((Img[i]-Min) / Divider)) + 1;
  If Color > 240 Then Color := 240;
  PlotPixel(Square[4].x,Square[4].y,Square[1].x,Square[1].y);
End;

Procedure PlotPixelColorOdd;
Begin
```

```
  Divider := (Max-Min) / 240;
  Color   := (Round((Img[i]-Min) / Divider)) + 1;
  If (Color > 240) And (Img[i] > Max-50) Then Color := 240;
  PutPixel(29+i,50+j,Color);
End;

Procedure PlotPixelColorEven;
Begin
  Divider := (Max-Min) / 240;
  Color   := (Round((Img[i]-Min) / Divider)) + 1;
  If (Color > 240) And (Img[i] > Max-50) Then Color := 240;
  PutPixel(30+h-i,50+j,Color);
End;

Procedure Plot_X;
Var Pos_Y,i,j : Integer;
Begin
  MousePos(Mouse); Pos_Y := Mouse.Y - 50;
  Reset(Fi,1); BlockRead(Fi,h,2);
  Rectangle(420,50,420+h,200);
  MaxDiv := (Max-Min) Div 35;
  For j := 1 To h-1 Do
   Begin
     BlockRead(Fi,Img,h*2);
     If j = Pos_Y Then
     Begin
       If Odd(j) Then
        Begin
          For i := 2 To h Do
           Begin
            SetViewPort(420,50,420+h,200,ClipOn);
```

```
            If Img[i] = 0 Then Img[i] := Min;
            PutPixel(i-1,150-((Img[i]-Min) Div MaxDiv),150);
            SetColor(150);
            If i <> 2 Then
            Line(i,150-((Img[i]-Min) Div MaxDiv),
                 i-1,150-((Img[i-1]-Min) Div MaxDiv));
           End;
          SetViewPort(0,0,GetMaxX,GetMaxY,ClipOn);
        End
        Else
        Begin
          For i := h DownTo 2 Do
           Begin
            SetViewPort(420,50,420+h,200,ClipOn);
            If Img[i] = 0 Then Img[i] := Min;
            PutPixel(h-i+1,150-((Img[i]-Min) Div MaxDiv),150);
            SetColor(150);
            If i <> h Then
            Line(h-i+1,150-((Img[i]-Min) Div MaxDiv),
                 h-i,150-((Img[i-1]-Min) Div MaxDiv));
           End;
          SetViewPort(0,0,GetMaxX,GetMaxY,ClipOn);
        End;
      End;
    End;
  SetColor(245);
  Close(Fi);
End;

Procedure Plot_Y;
Var Pos_X,i,j,ii : Integer;
```

```
    Data_Y : Array[1..360] Of Word;
Begin
  MousePos(Mouse); Pos_X := Mouse.X - 30; ii := 1;
  Reset(Fi,1); BlockRead(Fi,h,2);
  Rectangle(420,250,420+h,400);
  MaxDiv := (Max-Min) Div 35;
  For j := 1 To h-1 Do
   Begin
     BlockRead(Fi,Img,h*2);
     If Odd(j) Then
     Begin
      For i := 2 To h Do
       Begin
        If i = Pos_X Then
         Begin
           Data_Y[ii] := Img[i];
           ii := ii + 1;
         End;
       End;
     End
     Else
     Begin
       For i := h DownTo 2 Do
        Begin
          If i = h+1-Pos_X Then
           Begin
             Data_Y[ii] := Img[i];
             ii := ii + 1;
           End;
        End;
     End;
```

```
    End;
  Close(Fi);
  For ii := 1 To h-1 Do
    Begin
      If Data_Y[ii] > Max Then Data_Y[ii] := Min;
      SetViewPort(420,250,420+h,400,ClipOn);
      PutPixel(ii,150-((Data_Y[ii]-Min) Div MaxDiv),150);
      SetColor(150);
      If (ii <> 1) Then
      Line(ii,150-((Data_Y[ii]-Min) Div MaxDiv),
           ii-1,150-((Data_Y[ii-1]-Min) Div MaxDiv));
    End;
  SetViewPort(0,0,GetMaxX,GetMaxY,ClipOn);
End;

Begin
  SaveScreen(251,91,599,419);
  Frame3(251,91,599,419,9);
  SubTit(260,110,590,280,130,'Get Image File');
  SubTit(260,310,590,380,40,'Key');
  OutTxt(275,330,1,14,'[Esc]        [Spacebar]         ');
  OutTxt(275,330,1,15,'      - Quit            - Select');
  OutTxt(275,350,1,14,'[Up] & [Down] ');;
  OutTxt(275,350,1,15,'              Arrow - Select choice');
  OutTxt(275,130,1,14,'Path     :');
  OutTextXY(275,160,'Filename :');
  OutTextXY(275,190,'Method   : ');
  OutTxt(275,220,1,15,'( 1 - Mono  2 - Color  3 - Shade)');
  x := 390; y := 130; z := 0; BacG := 15; Index := 1;
  CurX := 390; CurY := 130; Created := 0; MethodStr := '';
  SetFillStyle(1,Bacg);
```

```
Bar(x-10,y-5,520,y+7);
SetColor(3);
GetDir(0,Path); OutTextXY(385,130,Path);
Repeat
  Ch := Readkey;
  If Ch = #0 Then
   Begin
     Ch := Readkey;
     Case Ch Of
     #72 : Begin
             OldY := y;  y := y - 30;
             Index := Index -1;
             If Index = 0  Then Index := 3;
             If Index = 3  Then y := 190;
             SetFillStyle(1,15); Bar(x-10,y-5,520,y+7);
             SetFillStyle(1,9);  Bar(x-10,OldY-5,520,OldY+7);
             SetFillStyle(1,15); WriteData1;
           End;
     #80 : Begin
             OldY := y; y := y + 30;
             Index := Index + 1;
             If Index = 4  Then Index := 1;
             If OldY = 190 Then y := 130;
             SetFillStyle(1,15); Bar(x-10,y-5,520,y+7);
             SetFillStyle(1,9);  Bar(x-10,OldY-5,520,OldY+7);
             SetFillStyle(1,15); WriteData1;
           End;
     End;
   End;
   If Ch = #32 Then
      Case Index Of
```

```
1 : Begin
     SetFillStyle(1,Bacg); Bar(x-10,y-5,520,y+7);
     CurX := 385; CurY := 130; SetColor(0);
     CursorOn; NoStr := 0; Path := '';
     Repeat
      Ch := Readkey;
      If (Ch = #8) And (NoStr <> 0) Then
       Begin
         CursorOff; CurX := CurX - TextWidth(Ch);
         Bar(CurX,CurY+5,CurX+7,CurY-3);
         CursorOn; Delete(Path,NoStr,1);
         NoStr := NoStr - 1;
       End;
      If (NoStr < 15) And (Ch <> #8) And (Ch <> #13)
         And (Ch <> #0) And (Ch <> #27) Then
       Begin
         CursorOff; OutTextXY(CurX,CurY-2,Ch);
         CurX := CurX + TextWidth(Ch);
         CursorOn; Path := Path + Ch;
         NoStr := NoStr + 1;
       End;
     Until Ch = #13;
     CursorOff; CurX := x; CurY := y; p := 1;
    End;
2 : Begin
       SetFillStyle(1,Bacg); Bar(x-10,y-5,520,y+7);
       CurX := 385; CurY := 160;
       InputName; Data := Sentence;
       Filename := Data + '.img';
    End;
3 : Begin
```

```
          Repeat
           SetFillStyle(1,BacG); Bar(x-10,y-5,520,y+7);
           CurX := 385; CurY := 190; InputNum;
           MethodStr := Num; Val(MethodStr,Method,Code);
          Until (Method > 0) And (Method < 4);
        End;
 End;  {End of Index}
If (Path <> '') And (Filename <> '') And (MethodStr <> '')
And (Ch = #13) Then
Begin
 If (Path = 'A:\') Or (Path = 'B:\') Or (Path = 'a:\') Or
    (Path = 'b:\') Then Delete(Path,3,1);
PathFile := Path + '\' + Filename;
FindFirst(PathFile,Archive,DirInfo);
Case DosError Of
  0 : Begin
        Max := 0; Min := $FFFF;
        Assign(Fi,Filename);
        {$I-}
        Reset(Fi,1);
        {$I+}
        If IOResult <> 0 Then
         Begin
           Halt(0);
           Writeln('Error opening image file : ',Filename);
         End;
        BlockRead(Fi,Header,2); h := Header;
        For j := 1 To h-1 Do
         Begin
           BlockRead(Fi,Img,h*2);
           For i := 1 To h Do
```

```
    Begin
      If Img[i] > Max Then Max := Img[i];
      If (Img[i] < Min) And (Img[i] <> 0) Then
       Begin
        Min := Img[i];
        OldMin := Min;
       End;
      If (Img[i] < Min) And (Img[i] = 0) Then
         Min := OldMin; '
    End;
 End;
BlockRead(Fi,Resol,2); Str(Resol,ResolStr);
 Case Resol Of
   10 : Begin
          Long := Round((h-1)/100);
          Str(Long,LongStr);
        End;
   20 : Begin
          Long := Round((h-1)/50);
          Str(Long,LongStr);
        End;
   40 : Begin
          Long := Round((h-1)/25);
          Str(Long,LongStr);
        End;
  100 : Begin
          Long := Round((h-1)/10);
          Str(Long,LongStr);
        End;
  200 : Begin
          Long := Round((h-1)/5);
```

```
          Str(Long,LongStr);
        End;
  250 : Begin
          Long := Round((h-1)/4);
          Str(Long,LongStr);
        End;
  500 : Begin
          Long := Round((h-1)/2);
          Str(Long,LongStr);
        End;
 1000 : Begin
          Long := Round(h-1);
          Str(Long,LongStr);
        End
 Else Begin
        Halt(0); Writeln('Error ! Error !');
      End;
 End;
Close(Fi);
Height := 25 * (Max-Min);
Str(Height,HeightStr);
Reset(Fi,1);
BlockRead(Fi,Header,2);
If (Method = 1) Or (Method = 2) Then
 Begin {*}
   OutTxt(295,250,1,14,'Ok !  Please, Wait !');
   Delay(1000);
   Frame3(272,115,578,280,9);
   OutTxt(275,130,1,14,'Origin (Y-Axis)  :');
   OutTxt(275,160,1,14,'Image Size (1-3) :');
   OutTxt(275,190,1,14,'Level (1-100)    :');
```

```
x := 440; y := 130;
CurX := 440; CurY := 130; Index := 1;
SetFillStyle(1,Bacg); Bar(x-10,y-5,520,y+7);
Data1 := ''; Data2 := ''; Data3 := '';
Repeat
 Ch := Readkey;
 If Ch = #0 Then
 Begin
  Ch := Readkey;
  Case Ch Of
  #72 : Begin
         OldY := y;  y := y - 30;
         Index := Index -1;
         If Index = 0  Then Index := 3;
         If Index = 3  Then y := 190;
         SetFillStyle(1,15);
         Bar(x-10,y-5,520,y+7);
         SetFillStyle(1,9);
         Bar(x-10,OldY-5,520,OldY+7);
         SetFillStyle(1,15); WriteData2;
        End;
 #80 : Begin
         OldY := y; y := y + 30;
         Index := Index + 1;
         If Index = 4  Then Index := 1;
         If OldY = 190 Then y := 130;
         SetFillStyle(1,15);
         Bar(x-10,y-5,520,y+7);
         SetFillStyle(1,9);
         Bar(x-10,OldY-5,520,OldY+7);
         SetFillStyle(1,15); WriteData2;
```

```
    End;
 End;{End Of Ch}
End;
If Ch = #32 Then
Case Index Of
 1 : Begin
      Repeat
       SetFillStyle(1,Bacg);
       Bar(x-10,y-5,520,y+7);
       CurX := 440; CurY := 130;
       InputNum; Data1 := Num;
       Val(Data1,YAxis,Code);
      Until (YAxis > 0) And (YAxis < 201);
     End;
 2 : Begin
      Repeat
       SetFillStyle(1,BacG);
       Bar(x-10,y-5,520,y+7);
       CurX := 440; CurY := 160;
       InputNum; Data2 := Num;
       Val(Data2,ImgSize,Code);
      Until (ImgSize > 0) And (ImgSize < 4);
     End;
3 : Begin
     Repeat
      SetFillStyle(1,BacG);
      Bar(x-10,y-5,520,y+7);
      CurX := 440; CurY := 190;
      InputNum; Data3 := Num;
      Val(Data3,Level,Code);
     Until (Level > 0) And (Level < 101);
```

```
      End;
    End; {End Of Index}
If (Data1 <> '') And (Data2 <> '') And (Data3 <> '') Then
 Begin
  MaxDiv := (Max-Min) Div Level;
  Created := 2;
  OutTxt(295,220,1,14,'Loading !  Loading !  Loading !');
  Delay(600);
  Closegraph; OpengraphSVGA;
  Case Method Of
    1 : Begin
          SetFillStyle(1,0); SetColor(11);
          For j := 1 To h-1 Do
           Begin
            BlockRead(Fi,Img,h*2);
            MoveTo(j*ImgSize,(j+YAxis)*ImgSize);
            Cancel;
            If (Odd(j)) And (Created = 2) Then
             Begint
               For i := 2 To h Do CreatOdd;
             End Else
            If Created = 2 Then
             Begin
               For i := h DownTo 2 Do CreatEven;
             End;
           End;
          Close(Fi);
          OutTxt(20,490,1,10,'Resolution      -');
          OutTextXY(20,510,'Size Of Image   -');
          OutTextXY(20,530,'Height Of Image -');
          OutTxt(170,490,1,14,ResolStr+' Microns.');
```

```
      OutTextXY(170,510,LongStr+' * '+LongStr+' mm.');
      OutTextXY(170,530,HeightStr+' Microns.');
      OutTextXY(10,560,'Press any key to goto Mainmenu !');
      Ch := Readkey; Ch := #27;
      Closegraph; Delay(200); Opengraph;
    End;
2 : Begin
      SetFillStyle(1,0); SetColor(Color);
      MixColor;
      For j := 1 To h-1 Do
       Begin
        SetColor(0);
        BlockRead(Fi,Img,h*2);
        MoveTo(j*ImgSize,(j+YAxis)*ImgSize);
        Cancel;
        If (Odd(j)) And (Created = 2) Then
         Begin
           For i := 2 To h Do
             Begin
               CreatOdd;
               PlotLineColor;
             End;
         End Else
        If Created = 2 Then
         Begin
           For i := h DownTo 2 Do
             Begin
               CreatEven;
               PlotLineColor;
             End;
         End;
```

```
          End;
          Close(Fi);
          For i := 1 To 240 Do
          Begin
            SetFillStyle(1,i); Bar(i*3,550,(i*3)+9,570);
          End;
          SetFillStyle(1,0); SetColor(150);
          OutTextXY(3,585,'Min ---------------------------------');
          OutTextXY(363,585,'---------------------------> Max');
          OutTxt(20,470,1,150,'Resolution      -');
          OutTextXY(20,490,'Size Of Image   -');
          OutTextXY(20,510,'Height Of Image -');
          OutTxt(170,470,1,190,ResolStr+' Microns.');
          OutTextXY(170,490,LongStr+' * '+LongStr+' mm.');
          OutTextXY(170,510,HeightStr+' Microns.');
          OutTextXY(20,535,'Press any key to goto Mainmenu !');
          Ch := Readkey; Ch := #27;
          Closegraph; Delay(200); Opengraph;
        End;
 End; {End Of Method}
End; {*}
  Until Ch = #27;
End Else
Begin
  Created := 2;
  OutTxt(295,250,1,14,'Loading !  Loading !  Loading !');
  Delay(600); Closegraph; OpengraphSVGA;
  If h > 361 Then
   Begin
     Closegraph; Delay(200);
     Writeln('FileSize too large ! Can'#39't execute.');
```

```
    Halt;
  End;
 If InstallMouse = False Then
  Begin
    Closegraph; Delay(200);
    Writeln('Mouse not install ! ');
    Halt;
  End
 Else
 Begin
  SetFillStyle(1,0); Bar(0,0,GetMaxX,GetMaxY);
  SetColor(245); Rectangle(535,420,770,500);
  Rectangle(420,420,515,500); Bar(430,408,462,425);
  Bar(545,408,575,425); MixColor;
  For i := 1 To 240 Do
  Begin
    SetFillStyle(1,i); Bar(i*3,550,(i*3)+9,570);
  End;
  SetColor(245); Rectangle(9,29,401,541);
  SetColor(150);
  OutTextXY(450,35,'Data On X-Axis');
  OutTextXY(450,235,'Data On Y-Axis');
  OutTextXY(550,417,'Key'); OutTextXY(435,417,'Pos');
  OutTextXY(440,440,'X : '); OutTextXY(440,470,'Y : ');
  OutTextXY(3,585,'Min ------------------------------------');
  OutTextXY(363,585,'----------------------------------> Max');
  OutTextXY(555,440,'Right Bottom - ');];
  OutTextXY(555,470,'Left  Bottom - ');
  OutTextXY(50,430,'Resolution      - ');
  OutTextXY(50,450,'Size Of Image   - ');
  OutTextXY(50,470,'Height Of Image - ');
```

```
SetColor(190);
OutTextXY(555,440,'                Plot Data');
OutTextXY(555,470,'                Move Mouse');
OutTextXY(200,430,ResolStr+' Microns.');
OutTextXY(200,450,LongStr+' * '+LongStr+' mm.');
OutTextXY(200,470,HeightStr+' Microns.');
Bar(30,490,380,530);
OutTextXY(70,505,'Click RightButtom Here To Escape !');
For j := 1 To h-1 Do
 Begin
   BlockRead(Fi,Img,h*2);
   If Odd(j) Then
     Begin
       For i := 2 To h Do PlotPixelColorOdd;
     End
   Else
     Begin
       For i := h DownTo 2 Do PlotPixelColorEven;
     End;
 End;
Close(Fi);
InitMouse1C; MouseShow1C;
ActiveArea(15,35,395,535);
Repeat
If RButtonPress And InArea(31,51,31+h-2,51+h-2) Then
 Begin
   Repeat Until AllButtonRelease;
   Plot_X; Plot_Y;
   MouX := Mouse.X-30; MouY := Mouse.Y-50;
   Str(MouX,Pos_MouseX); Str(MouY,Pos_MouseY);
   SetColor(190);
```

```
                OutTextXY(470,440,Pos_MouseX);
                OutTextXY(470,470,Pos_MouseY);
                SetColor(245);
                Repeat
                  SetMousePos(Mouse.X,Mouse.Y);
                Until LButtonPress;
                Repeat Until AllButtonRelease;
                SetFillStyle(1,0);
                Bar(420,50,420+h,200); Bar(420,250,420+h,400);
                Bar(469,435,510,480);
              End;
             Until RButtonPress And InArea(30,490,380,530);
             Delay(200); Ch := #27;
             MouseHide1C; Closegraph; Delay(200); Opengraph;
            End;
          End;
        End; {End Of 0}
    3 : Begin
          OutTxt(295,250,1,14,'Path Not Found ! Try Again.');
          Ch := Readkey;
          SetFillStyle(1,9); Bar(290,245,510,260);
        End;
  2,18 : Begin
          OutTxt(295,250,1,14,'File Not Found ! Try Again.');
          Ch := Readkey;
          SetFillStyle(1,9); Bar(290,245,510,260);
        End;
    Else Begin
          OutTxt(295,250,1,14,'Error ! Error ! Error !');
          Ch := Readkey;
          SetFillStyle(1,9); Bar(290,245,510,260);
```

```
            End;
         End;
       End;
    If Ch = #27 Then
       Case Created Of
         0 : Begin
              RestoreScreen(251,91);
              FreeMem(Pt,ImageSize(251,91,599,419));
             End;
       1,2 : Begin
              Title1; Date;
              RestoreScreen(251,91);
              FreeMem(Pt,ImageSize(251,91,599,419));
              i := 1; x := 75; y := 150; ForG := 15; BacG := 3;
              SetFillStyle(1,BacG); Bar(65,195,190,210);
              For i := 1 To MaxMenu Do
               Begin
                 CtlText(x,y,ForG,BacG,Item[i]); y := y + 50;
               End;
             End;
       End;
  Until Ch = #27;
  x := 75; y := 200;
End;

         {********** End of Get Image ********}

       {********** Control Stepping Motor *********}

Procedure CtMotor;
Var Step    : Integer;
```

```
    StepStr : String;

Procedure Cw200Step;
Var n : Integer;
Begin
  For n := 1 To Step Do
   Begin
     Port[PA1] := Data_Out[CwIndex];
     Delay(5); Inc(CwIndex);
     If CwIndex > 4 Then CwIndex := 1;
   End;
End;

Procedure Ccw200Step;
Var n : Integer;
Begin
  For n := 1 To Step Do
   Begin
     Port[PA1] := Data_Out[CcwIndex];
     Delay(5); Inc(CcwIndex);
     If CcwIndex > 8 Then CcwIndex := 5;
    End;
End;

Procedure YCw200Step;
Var n : Integer;
Begin
  For n := 1 To Step Do
   Begin
     Port[PA1] := Data_Out[YcwIndex];
     Delay(5); Inc(YcwIndex);
```

```
    If YcwIndex > 12 Then YcwIndex := 9;
   End;
End;

Procedure YCcw200Step;
Var n : Integer;
Begin
  For n := 1 To Step Do
   Begin
     Port[PA1] := Data_Out[YccwIndex];
     Delay(5); Inc(YccwIndex);
     If YccwIndex > 16 Then YccwIndex := 13;
   End;
End;

Procedure Buttom(x1,y1,x2,y2 : Integer; Msg : String);
Begin
  SetFillStyle(1,9); Bar(x1,y1,x2+7,y2+7);
  SetFillStyle(1,0); Bar(x1+7,y1+7,x2+7,y2+7);
  SetFillStyle(1,7); Bar(x1,y1,x2,y2);
  OutTextXY(x1+5,y1+5,Msg);
End;

Procedure Up;    Begin  Buttom(380,130,420,150,' Up');   End;
Procedure Left;  Begin  Buttom(330,165,370,185,'Left');  End;
Procedure Right; Begin  Buttom(430,165,480,185,'Right'); End;
Procedure Down;  Begin  Buttom(380,200,420,220,'Down '); End;

Procedure ButtomPress(x1,y1,x2,y2 : Integer; Msg : String);
Begin
  SetColor(15);
```

```
  SetFillStyle(1,9); Bar(x1-5,y1-7,x2+7,y2+7);
  SetFillStyle(1,7); Bar(x1,y1,x2,y2);
  OutTextXY(x1+5,y1+5,Msg); Delay(500);
End;

Procedure UpPr; Begin ButtomPress(385,135,425,155,' Up');  Up;    End;
Procedure LtPr; Begin ButtomPress(335,170,375,190,'Left'); Left;  End;
Procedure RtPr; Begin ButtomPress(435,170,485,190,'Right');Right; End;
Procedure DnPr; Begin ButtomPress(385,205,425,225,'Down'); Down;  End;

Begin
  SaveScreen(251,91,599,419);
  Frame3(251,91,599,419,9);
  SubTit(260,110,590,270,195,'Control Stepping Motor');
  Up; Down; Left; Right; SubTit(260,310,590,400,40,'Key');
  OutTxt(275,330,1,14,'[Arrow Key]');
  OutTxt(275,330,1,15,'            - Control Motor');
  OutTxt(275,350,1,14,'[Esc]');
  OutTxt(275,350,1,15,'      - End and Clear data');
  OutTxt(275,370,1,14,'[S Or s] -');
  OutTxt(360,370,1,15,' Change Step');
  OutTxt(290,250,1,14,'Step - ');
  OutTxt(410,250,1,14,'(200,400 Or 600 Only)');
  SetFillStyle(1,15); Bar(345,245,390,258);
  Step := 400; Str(Step,StepStr);
  OutTxt(355,250,1,0,StepStr); SetColor(15);
  Port[PCC1] := $80; Mode := 0;
   Repeat
     Ch := Readkey;
       If Ch = #0 Then
         Begin
```

```
Ch := Readkey;
Case Ch Of
  #75 : Begin
          LtPr;
          If Mode <> 1 Then
            Begin
              Mode :=1; CwIndex := 1;
            End;
          Cw200Step;
        End;
  #77 : Begin
          RtPr;
          If Mode <> 2 Then
            Begin
               Mode := 2; CcwIndex := 5;
            End;
          Ccw200Step;
        End;
  #72 : Begin
          UpPr;
          If Mode <> 3 Then
            Begin
              Mode := 3; YcwIndex := 9;
            End;
          YCw200Step;
        End;
  #80 : Begin
          DnPr;
          If Mode <> 4 Then
            Begin
              Mode := 4; YccwIndex := 13;
```

```
                End;
              YCcw200Step;
            End;
        End;
      End;
    If UpCase(Ch) = 'S' Then
      Begin
        Repeat
         SetFillStyle(1,15);
         Bar(345,245,390,258); CurX := 355; CurY := 250;
         InputNum; StepStr := Num; Val(StepStr,Step,Code);
        Until (Step = 200) Or (Step = 400) Or (Step = 600);
      End;
  Until Ch = #27;
  Port[PA1] := $00;
  RestoreScreen(251,91);
  FreeMem(Pt,ImageSize(251,91,599,419));
End;

        {********** End of CtMotor ***********}

        {************ Information ***********}

Procedure Information;
Var  U : Integer;
Begin
 Title1; U := 0;
 Frame0(35,60,600,420,9);
 SubTit(50,90,590,260,250,'LASER SURFACE SCANNING SYSTEM');
 OutTxt(110,110,1,15,'The application of laser is utilized in serveral aspects.');
 OutTextXY(70,130,'This research presents surface image processing by using LASER ');
```

```
 OutTextXY(70,150,'application.  The device, Laser Displacement Sensor (LDS), can');
 OutTextXY(70,170,'detect depth of material surface  and collect data by scanning');
 OutTextXY(70,190,'through interfacing circuit to Microcomputer for processing the');
 OutTextXY(70,210,'image of the object surface by using Turbo Pascal programming. ');
 OutTextXY(70,230,'This result can process image in certain in a level.');
 SubTit(50,275,310,320,45,'Key');
 OutTxt(70,290,1,14,'[Esc]');
 OutTxt(70,290,1,15,'      - To Goto Mainmenu.');
   Repeat
    Ch := Readkey;
    If (Ch = #27) Then
     Begin
       Title1; Describe4; Date; Key;
       x := 75; y := 150; ForG := 15; BacG := 3;
       SetFillStyle(1,BacG); Bar(65,295,190,310);
       For i := 1 To MaxMenu Do
        Begin
          CtlText(x,y,Forg,Bacg,Item[i]); y := y + 50;
        End;
     End;
   Until (Ch = #27);
End;

      {************ End of Information *************}

         {************** Quit *************}

Procedure Quit;
Begin
  SaveScreen(240,200,440,280);
  Frame1(240,200,440,280,4);
```

```
  OutTxt(278,220,1,15,' EXIT PROGRAM ');
  OutTxt(260,240,1,15,'Are You Sure ? (Y/N)');
  Repeat
   Ch := Readkey;
   If UpCase(Ch) = 'Y' Then
    Begin
      Check := True; Exit;
    End;
   If UpCase(Ch) = 'N' Then
    Begin
      RestoreScreen(240,200);
      FreeMem(Pt,ImageSize(240,200,440,280));
      i := 5;
    End
  Until (UpCase(Ch) = 'Y') Or (UpCase(Ch) = 'N');
End;

          {************ End of Quit ***************}

            {****************MAIN********************}

Begin
 ClrScr; Opengraph;
 Title; Title1; Date;
 Check := False; Index := 1; Describe1; Key;
 x := 75; y := 150; z := 0; ForG := 15; BacG := 3;
 SetFillStyle(1,Bacg); Bar(x-10,y-5,190,y+10);
 For i := 1 To MaxMenu Do
  Begin
    CtlText(x,y,Forg,Bacg,Item[i]); y := y + 50;
  End;
```

```
  Repeat
    Forg := 15; BacG := 3;
    A := PopUp(x,y,ForG,BacG,Item);
    Case A Of
      0 : Quit;
      1 : Scan;
      2 : Get_Image;
      3 : CtMotor;
      4 : Information;
    End;
  Until Check = True; Delay(500);
 Closegraph; Delay(500);
 Writeln('CopyRight (c) Applied Physics KMIT'#39'L 1993-1994. ');
 Writeln('Thanks !! See You Again. ');
End.
```

ภาคผนวก ข.

Data Sheet

KEYENCE

LB-70
LB-72

Laser displacement sensor

INSTRUCTION MANUAL

CONTENTS

Thank you for choosing a Keyence LB series laser displacement sensor.
KEYENCE has developed Hi-Tech Sensors incorporating up-to-the-minute technology to meet our customers' needs for factory automation. We have gained an excellent reputation as a leading manufacturer of industrial sensors, for both our advanced technology and years of experience.
Our careful quality management, strict inspection, and complete after-sales service have resulted in high customer satisfaction and great confidence in our products.
Please read this manual thoroughly to ensure correct operation and to derive maximum benefit from this device.

SAFETY 3
PART NAMES 6
CONNECTIONS 7
INSTALLATION 8
ADJUSTMENT 9, 10
TROUBLESHOOTING 11
CHARACTERISTIC DATA 12, 13
SPECIFICATIONS 14, 15
DIMENSIONS 16
HINTS ON CORRECT USE 17

Features

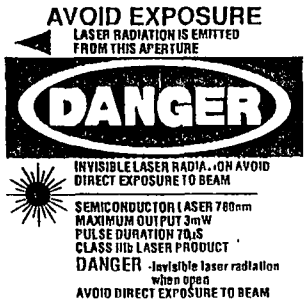
- Extremely wide measurement range:
  60 mm to 140 mm: LB-70
  30 mm to 50 mm: LB-72
- High response speed of
  0.7 ms (700 Hz): LB-70
  0.15 ms (3 kHz): LB-72
- High resolution of
  10 μm: LB-70 (Object: white paper at a distance of 100 mm, at 500 ms)
  2 μm: LB-72 (Object: white paper at a distance of 40 mm, at 60 ms)
- Zero point adjustment in the full measurement range
  60 mm to 140 mm: LB-70
  30 mm to 50 mm: LB-72
- Compact body can be attached with DIN rail
- Highly flexible reinforced cables (LB-11)

SAFETY

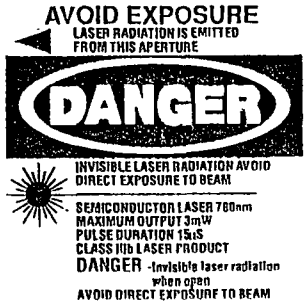
The laser displacement sensor LB series is classified as a Class IIIb laser product, and is identified as such by the warning logotype shown in the figure below, which also shows other warning labels affixed to the covers. If the operator stares directly into the beam, damage can be caused to the operator's eyes.

Drawings

LB-70/LB-11



LB-72/LB-12



Safety precautions:

The LB series uses a semiconductor laser of 780nm wavelength as its light source. It is activated in a pulse form and the peak power of this laser from the aperture is 3mW max (LB-70), 1.5mW max (LB-72).

The laser beam will not cause any sort of damage to the operator's skin. There is no danger in exposing arms or hands to the laser beam. The only possible health hazard is exposure of the eyes to the laser beam. Damage to the eyes can occur if the operator stares directly into the laser beam.

Following the safety precautions below will assure safety for the operator:

1. Do not stare directly into the laser beam.
2. Exercise caution when detecting extremely reflective parts. Do not stare into the reflected laser light.
3. Observe all warning labels.
4. CAUTION: Use of controls or adjustments or performance of procedures other than those specified herein may result in hazardous radiation exposure.
5. Do not attempt to open the unit for repair. A defective unit has to be returned to the manufacturer for repair.
6. All laser products must conform to the requirements of 21 CFR Subchapter J. This federal regulation is administered by the Center for Devices and Radiological Health (CDRH) under the Food and Drug Administration.
7. It is necessary for the original equipment manufacturers (OEMs) to certify that the products or system which the OEMs sell to end users conforms to all pertinent CDRH requirements.

For further information,.contact: Center for Devices and
Radiological Health
(HFZ-300)
Rockville, Maryland 20857

Laser safety precautions

1) The LB series is provided with the following safety features. Make sure these features function correctly prior to making any measurement.

I LASER ON alarm LED
A visible LED that informs you that the laser beam is being emitted or is about to be emitted at least 3 sec. after power is provided to the control panel and the sensor head.

II Delaying laser emission
The laser emission starts more than 3 seconds after the LED lights, thus decreasing the possibility of laser exposure.

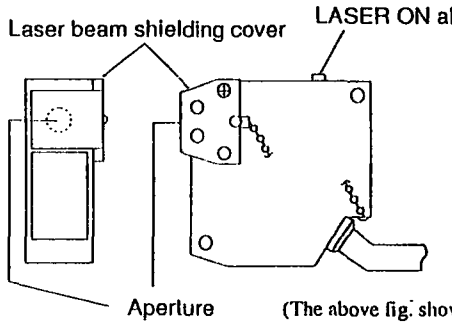
III Laser emitting control remote terminal
A remote terminal for controlling laser emission is provided on the front of the controller. You can remotely control laser emission using this terminal.

IV Key power switch
Set to the ON position for supplying power.
You can lock the controller power switch using the supplied key, which can be removed only when set to the OFF position.

V Laser beam shielding cover
A laser beam shielding cover is supplied. This cover is to be attached to the laser-beam-emitting side of the sensor head. If an operator must work in front of the sensor head and there is risk to the eyes from the laser beam, be sure to attach this cover.

When the shielding cover is used:

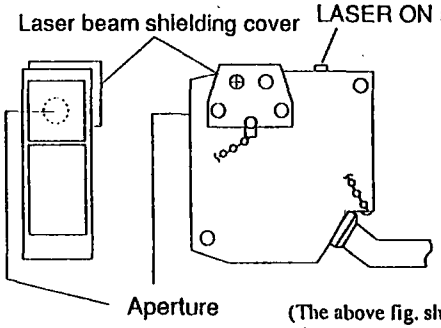
Attach the shielding cover to the front surface of the light source lens and secure it using the supplied screw.



(The above fig. shows the appearance of the LB-12)

When the shielding cover is not used:

Secure the shielding cover to the screw hole on the rear of the sensor head using the supplied screw.



(The above fig. shows the appearance of the LB-12)

2) Warning indicator panel
When using the sensor head, install a warning indicator panel on which a warning appropriate for Class IIIb is displayed.
* Install the panel in the vicinity of the measurement location so that the panel is not exposed to the laser beam.

3) Prevention of laser beam reflection
It is preferable to place a material having high light absorbency and scattering power at the end of the laser beam path. If a reflective material such as a mirror is placed at the end of the laser beam path, the laser beam may be reflected into your eyes.

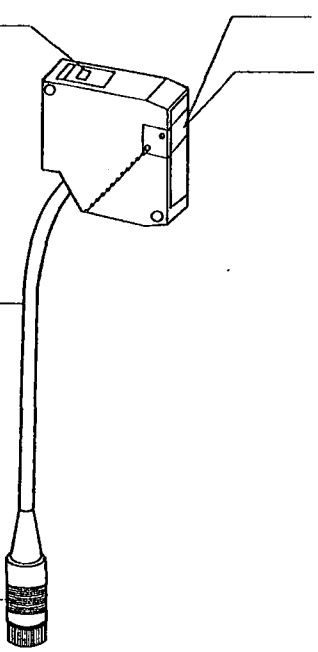
4) Disassembly of sensor head
Never disassemble the sensor head.
The sensor head is not provided with a function that automatically stops laser beam emission when the sensor head is disassembled. Therefore, if you should disassemble the sensor head, you may be exposed to laser beam emission.

5) Warning
Strictly observe the cautions described in this manual, otherwise you may be subjected to risk of laser beam exposure.

PART NAMES

● Sensor head

LB-11 (Wide measurement range model)
LB-12 (High resolution model)



LASER ON alarm/ STABILITY LED Indicator
- Indicates the measurement range by changing colors.
- Indicates, by changing colors, when the light quantity exceeds or falls below a specific limit. (See P4.)

Sensor cable (length: 2 m)
- Made of high-tension copper wire that retains high flexibility. (LB-11)

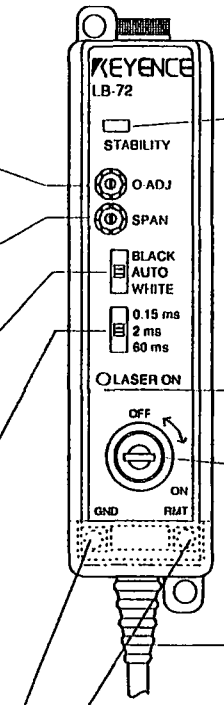
Connector

Laser-emitting portion

Laser beam shield cover (See P4.)

● Controller

LB-70 (Wide measurement range model)
LB-72 (High resolution model)



0-ADJ adjuster trimmer
- Adjusts the reference voltage (zero point of the analog output voltage) to any point within the measurement range

SPAN adjuster trimmer
- Adjusts the inclination of the output voltage curve.

Sensitivity selector switch
- In addition to BLACK or WHITE settings, there is an automatic gain circuit that increases the sensitivity by reducing the effect of the color and material of objects.

Response speed selector switch
- Selects one of three response speeds:
  0.15 ms, 2 ms, and 60 ms (LB-72).
  0.7 ms, 20 ms, and 500 ms (LB-70).

STABILITY LED Indicator
- Indicates the measurement range by changing colors.
- Indicates, by changing colors, when the light quantity exceeds or falls below a specific limit.

LASER ON alarm LED Indicator (See P4.)

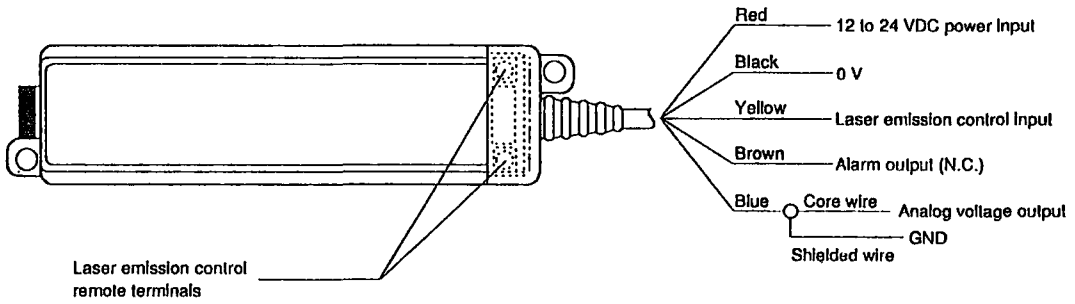
Key power switch (See P4.)

I/O cable (length: 2 m)

Laser emitting control remote terminals (See P4.)

(The above fig. shows the appearance of the LB-72.)

CONNECTIONS



● Laser emitting control input

When the yellow and black cables are short-circuited, the laser beam stops emitting temporally.
Analog output voltage just before the short-circuiting will be freezed.
This input is used when emitting laser beams alternatively to prevent mutual interference, or when turning off laser beams in an emergency.

● Alarm output (N.C.)

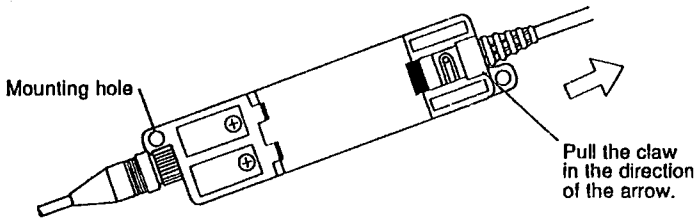
Outputs when the light quantity exceeds or falls below a specified limit.

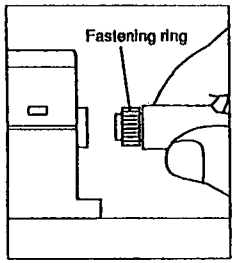
● Laser emitting control remote terminals

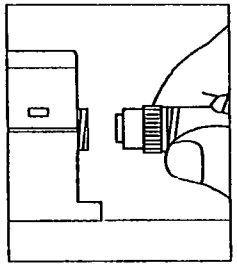
When the terminals are short-circuited, the LASER ON LED on the front panel lights and laser emission starts more than 3 sec. later.
When the terminals are opened, the LASER ON LED on the front panel goes out and laser emission stops.
* When shipped, a short-circuit bar is inserted between terminals.

INSTALLATION

① Controllers can be attached on DIN rails. Pull the claw at the bottom in the direction of the arrow when installing or removing controllers.







② LB-70

Turn the fastening ring to connect the connectors. Turning the cable may cause disconnection.

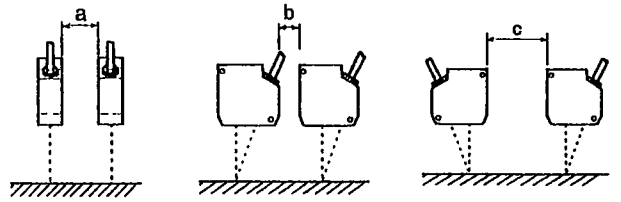
LB-72

When attaching the connector, gently press the plug into the receptacle and rotate the plug right and left to locate the engaging position. Insert the plug until you hear a clicking sound.

When removing the connector, hold the connection sleeve as shown in the fig. on the right and pull the connector out in the direction indicated by the arrow.

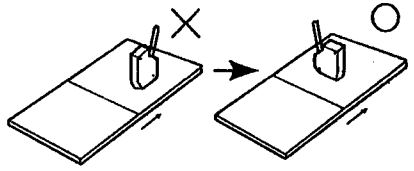
③ Confirm the mutual interference area below when using sensor heads side by side.

Mutual interference area



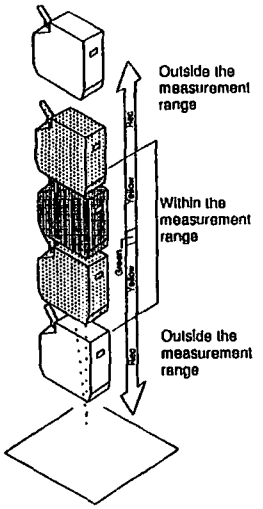
	a	b	c
LB-70	65 mm	50 mm	90 mm
LB-72	45 mm	20 mm	35 mm

When an object to be measured consists of differently colored portions or different materials separated by a border line, the measurement result may vary depending on the installation direction of the sensor. To minimize deviation of the measurement, install the sensor parallel to the border line as shown in the fig. below.

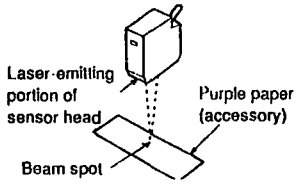


ADJUSTMENT

① Securely install the sensor in desired location, 60 mm to 140 mm (LB-70) or 30 mm to 50 mm (LB-72) from the object. Note that the laser-emitting portion of the sensor is parallel to the object. When the object is within the measurement range, the STABILITY LED indicator turns yellow. When the object is at the center of the measurement range, about 100 mm (LB-70) or 40 mm (LB-72) away from the sensor, the STABILITY LED indicator turns green.

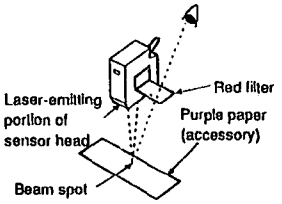


② Put the purple paper (accessory) correctly below the laser-emitting portion of the sensor head to fix the location of the beam spot. Temporarily fix the sensor head at that position.



※ Put the beam spot area in shadow to make confirmation easier.

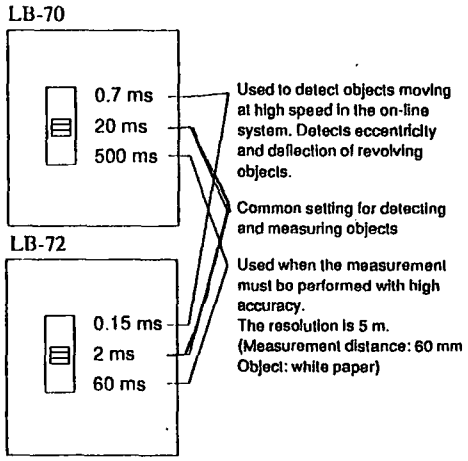
③ After confirming the location of the beam spot, fold the red filter (accessory) and attach to the sensor head.



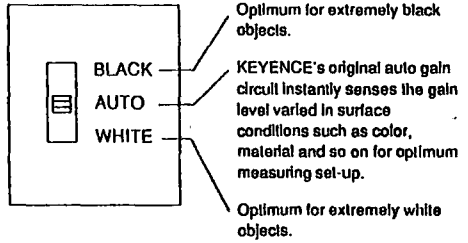
④ Confirm the beam spot on the purple paper through the red filter. Tighten the fastening screw of the sensor head.

※ After tightening the fastening screw, remove and keep the purple paper and the red filter.

⑤ Select the response speed using the response speed selector switch according to the speed of the object.

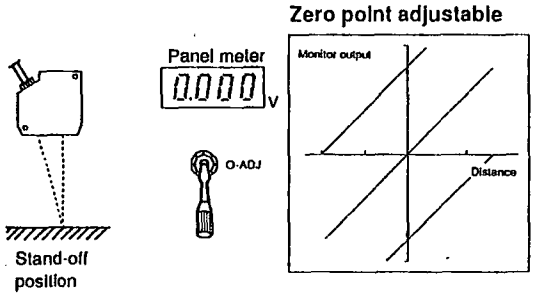


⑥ Select the appropriate sensitivity using the sensitivity selector switch.

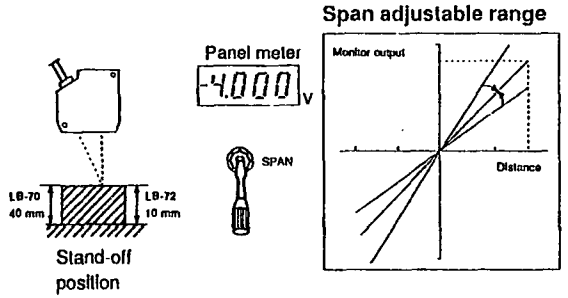


ADJUSTMENT

① When the object to be detected is displaced from the reference position, the analog output voltage may deviate from the zero point. In such a case, the 0-ADJ adjuster can adjust the zero point.



② Confirm the value of the output voltage to see whether the changes in output voltage are proportional to the displacement of the object. If not, adjust the output voltage by using the SPAN adjuster.



③ Confirm that the analog output voltage is 0 V at the reference position. If not, repeat the above adjustment procedures.

TROUBLESHOOTING

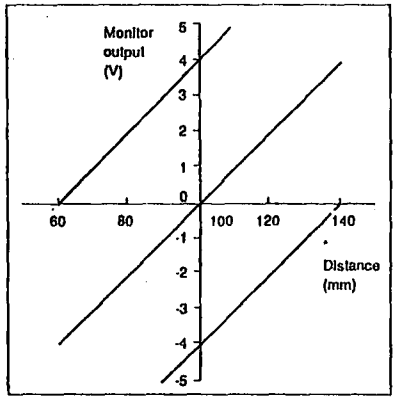
When abnormal operation occurs, first check following, Trouble-shooting points.
If abnormal operation continues after taking the steps outlined below, please contact our nearest distributor or the Overseas Division of Keyence Corporation.

Troubles	Item to be checked	Countermeasures	Reference
The monitor voltage does not change.	Is the sensor head installed in the correct position?	Install the sensor head in the range where the STABILITY LED indicator turns yellow or green.	P9
	Is the connector of the sensor head tightened firmly?	Firmly attach the connector by completely turning the fastening ring.	P8
The resolution is not high enough.	Does the STABILITY LED indicator turn red?	Adjust the position and angle of the sensor until the STABILITY LED indicator turns yellow or green.	P9
	Is the voltage in proportion to the displacement output?	Adjust the output voltage using the SPAN dial.	P10
	Is the response speed selector switch set to 0.7 ms? 0.15 ms?	Set the response speed selector switch to SLOW when fast response is not required.	P9
	Is the sensor affected by external disturbance light?	Take countermeasures against external disturbance light so that it does not fall on the lenses of the sensor.	P17
	Is the main power voltage at the correct level?	Use a stabilized power supply* so that the power voltage stays within the allowable range.	—
The monitor voltage fluctuates.	Are the lenses of the sensor head clean?	Wipe off dirt or dust with a soft cloth.	P17
	Is the sensor affected by noise?	Route the wiring of the sensor away from power cables and high-tension cables.	P17
	Is there any vibration on the feeder line?	Reduce vibration near the measurement point as much as possible.	—
	Is the main power voltage at the correct level?	Use a stabilized power supply* so that the power voltage stays within the allowable range.	—
The monitor voltage changes drastically when two sensor heads are used, e.g. when measuring thickness.	Is the span range of only one sensor adjusted?	Adjust the span range of both sensors.	—

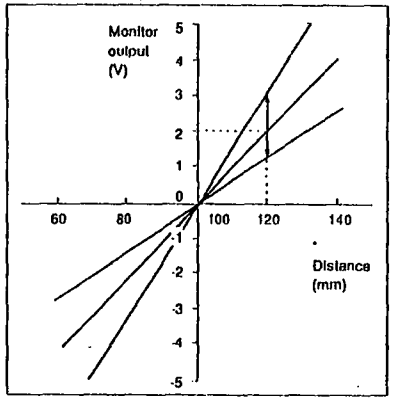
※ A stabilized DC power supply (CU series) is available. Contact KEYENCE for details.

CHARACTERISTIC DATA: LB-70

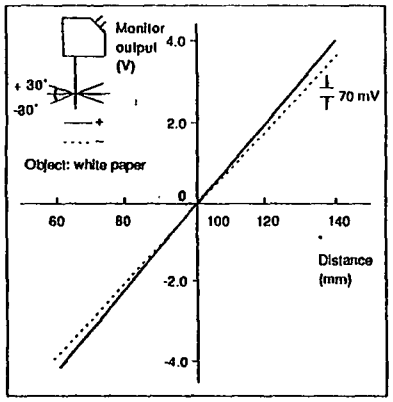
Zero point adjustable range

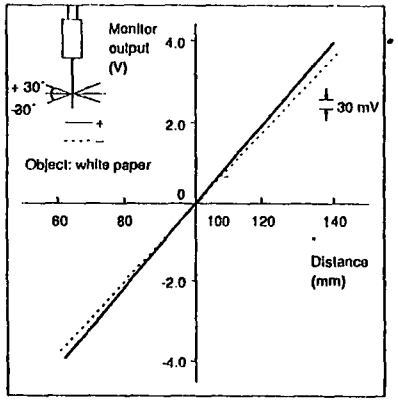


Span adjustable range



Changes in detection span when the object to be detected is tilted

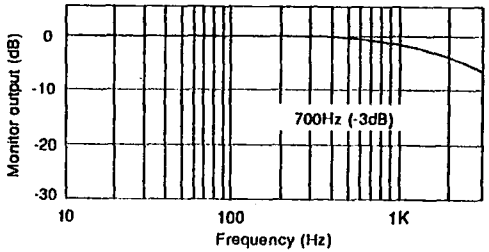




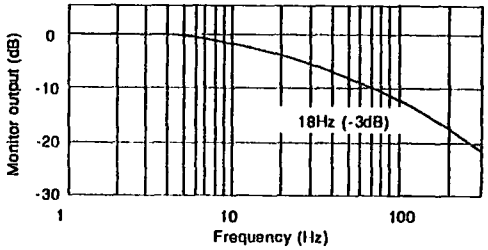
※ Distance between the laser-emitting portion of the sensor head and the object

Frequency characteristics

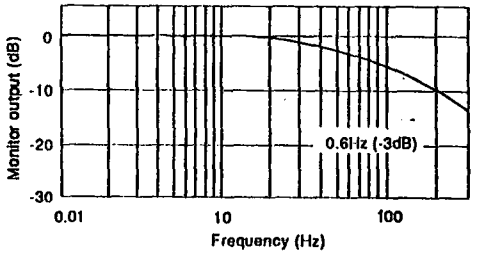
The response speed selector switch is set to 0.7 ms.



The response speed selector switch is set to 20 ms.

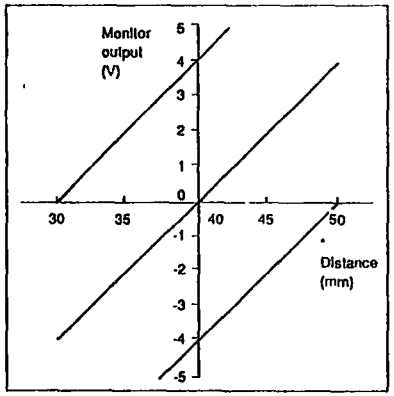


The response speed selector switch is set to 500 ms.

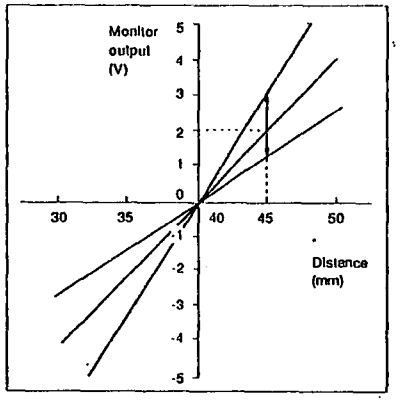


CHARACTERISTIC DATA: LB-72

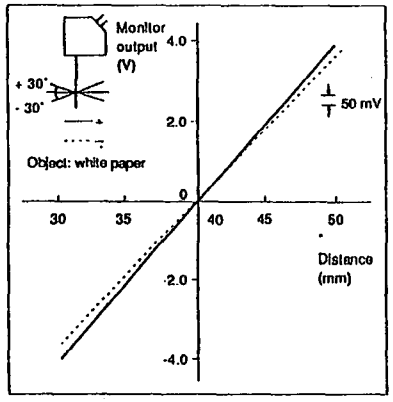
Zero point adjustable range

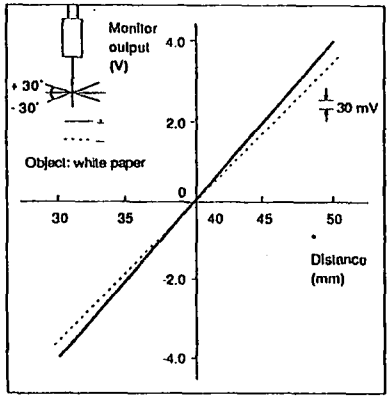


Span adjustable range



Changes in detection span when the object to be detected is tilted

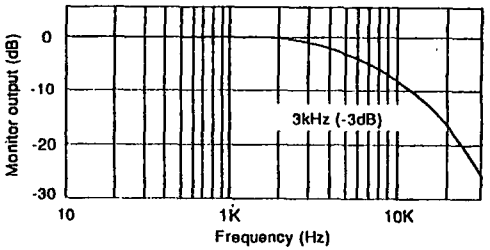




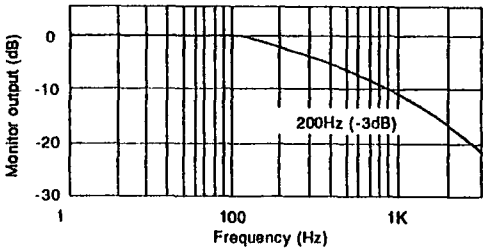
* Distance between the laser-emitting portion of the sensor head and the object

Frequency characteristics

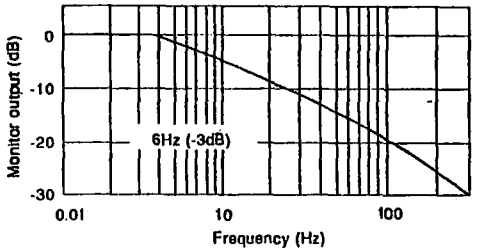
The response speed selector switch is set to 0.15 ms.



The response speed selector switch is set to 2 ms.



The response speed selector switch is set to 60 ms.



SPECIFICATIONS: LB-70

Model	Sensor head		LB-11
	Controller		LB-70
Stand-off distance*1			100 mm
Measurement range			±40 mm
Laser			Semiconductor laser (wavelength: 780 nm) max. 3 mW, CLASS IIIb Pulse duration: max. 70 µs
Spot diameter*2			1.0 x 2.0 mm
Linearity*5			1.6% of F.S.
Resolution*3			10 µm (500 ms)/40µm (20 ms)/180 µm (0.7 ms)
STABILITY LED indicator			LED (3 colors)
Output	Analog	Output voltage*4	–4 V to +4 V (0.1 V/mm)
		Output impedance	100Ω
	Alarm output		NPN open collector: 50 mA (40 V) max. Residual voltage: 1 V max. (N.C.)
Zero point adjustment range			60 mm to 140 mm
Span adjustment range			0.1 V/mm±30%
Response speed			DC–700Hz (at 0.7 ms) –3 dB/DC–18Hz (at 20 ms) –3 dB/DC–0.6Hz (at 500 ms) –3dB
Sensitivity			WHITE, BLACK, and AUTO (selector switch)
Temperature fluctuation*5		Sensor head	0.02% of F.S./°C
		Controller	0.04% of F.S./°C
Timing input			NPN open collector or non-voltage contact output
Operating illumination*6			4000 lx max.
Operating temperature			0°C to +50°C
Operating humidity			35% to 85%RH (without condensation)
Supply voltage			12 to 24 VDC ± 10% Ripple (P-P): 10% max.
Power consumption			120 mA MAX.
Material	Sensor head		Die cast zinc-based alloy
	Controller		Polycarbonate
Weight	Sensor head		Approx. 180 g (including cable)
	Controller		Approx. 185 g (including cable)

*1 Distance from the surface of the laser-emitting portion of the sensor head.
*2 The beam spot is visible when the object (white paper) is placed at a distance of 100 mm from the laser-emitting portion of the sensor head.
*3 The value when the object (white paper) is measured at a distance of 100 mm from the laser-emitting portion of the sensor head.
*4 The range can be expanded to "–5 V to +5 V" by using the 0-ADJ and SPAN adjuster trimmers.
*5 F.S. stands for a measurement range of 80 mm.
*6 Incandescent lamp or fluorescent lamp

SPECIFICATIONS: LB-72

Model	Sensor head		LB-12
	Controller		LB-72
Stand-off distance			40 mm
Measurement range*1			±10 mm
Laser			Semiconductor laser (wavelength: 780 nm) max. 3 mW, CLASS IIIb Pulse duration: max. 15 µs
Spot diameter*2			1.0 mm
Linearity*5			1% of F.S.
Resolution*3			2 µm (60 ms)/15µm (2 ms)/50 µm (0.15 ms)
STABILITY LED indicator			LED (3 colors)
Output	Analog	Output voltage*4	–4 V to +4 V (0.4 V/mm)
		Output impedance	100Ω
	Alarm output		NPN open collector: 50 mA (40 V) max. Residual voltage: 1 V max. (N.C.)
Zero point adjustment range			30 mm to 50 mm
Span adjustment range			0.4V/mm±30%
Response speed			DC~3KHz (at 0.15 ms) –3 dB/DC~200Hz (at 2 ms) –3 dB/DC~6Hz (at 60 ms)
Sensitivity			WHITE, BLACK, and AUTO (selector switch)
Temperature fluctuation*5		Sensor head	0.04% of F.S./°C
		Controller	0.03% of F.S./°C
Timing input			NPN open collector or non-voltage contact output
Operating illumination*6			4000 lx max.
Operating temperature			0°C to +50°C
Operating humidity			35% to 85%RH (without condensation)
Supply voltage			12 to 24 VDC ± 10% Ripple (P-P): 10% max.
Power consumption			120 mA MAX.
Material	Sensor head		Die cast zinc-based alloy
	Controller		Polycarbonate
Weight	Sensor head		Approx. 250 g (including cable)
	Controller		Approx. 160 g (including cable)

*1 Distance from the surface of the laser-emitting portion of the sensor head.
*2 The beam spot is visible when the object (white paper) is placed at a distance of 40 mm from the laser emitting portion of the sensor head.
*3 The value when the object (white paper) is measured at a distance of 40 mm from the laser-emitting portion of the sensor head.
*4 The range can be expanded to "–5 V to +5 V" by using the 0-ADJ and SPAN adjuster trimmers.
*5 F.S. stands for a measurement range of 20 mm.
*6 Incandescent lamp or fluorescent lamp

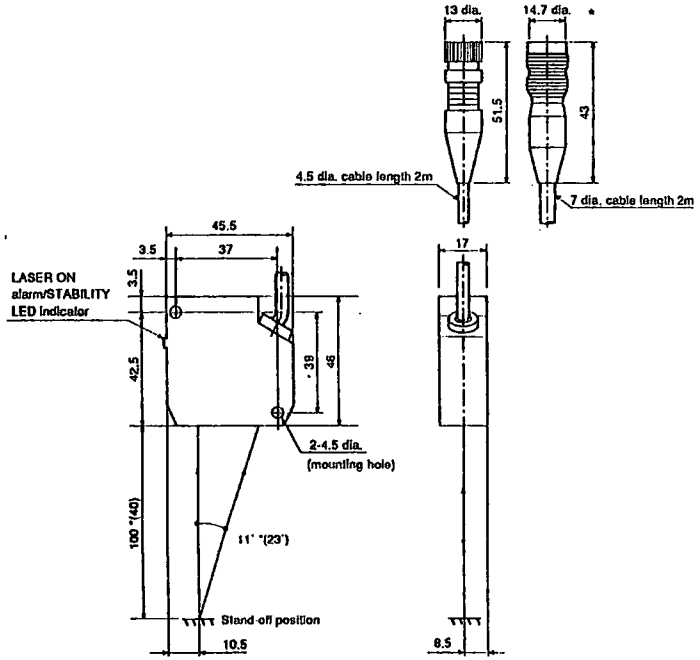
DIMENSIONS

[Unit: mm]

Sensor

LB-11

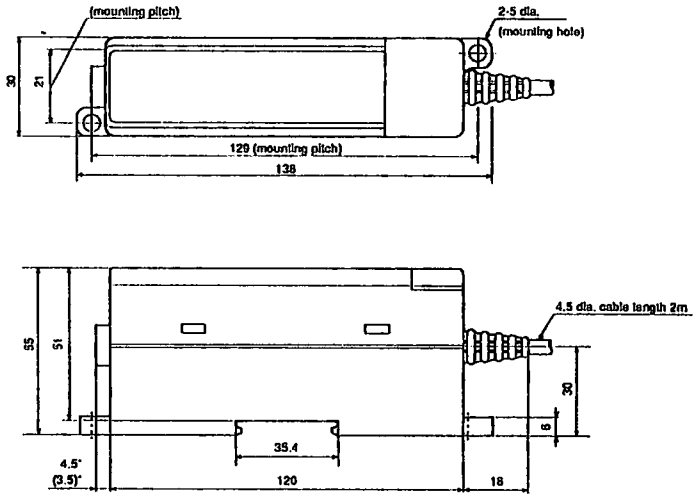
LB-12 Items marked with * are different from those of LB-11.



Controller

LB-70

LB-72 Items marked * are different from those of LB-70.

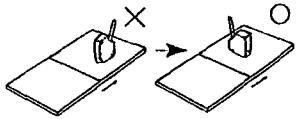


HINTS ON CORRECT USE

■ Sensor Mounting

When an object to be measured consists of differently colored portions or different materials separated by a border line, the measurement result may vary depending on the installation direction of the sensor. To minimize deviation of the measurement, install the sensor parallel to the border line as shown in the fig. below.

- Completely turn the mounting ring when installing the sensor head. Avoid turning the cable when installing or removing the sensor; it may cause disconnection.



■ Object to Be Detected

- Mirror-surfaced objects may not be detected depending on the direction in which the light beam is reflected from the object. Carefully adjust the angle of the sensor head to the object.

■ Compatibility

The sensor and the controller are calibrated in pairs at the factory. Be sure to use a sensor and controller having the same serial number.

■ Sensor Cable

Route the wiring of the sensor away from power cables and high-tension cables, as these may cause the sensor to malfunction. The maximum cable length is 10 m.

An extension connector cable is available as an option. Contact KEYENCE for details.

■ Atmosphere

- Always keep the sensor clean so that no matter that can reflect the light beam (such as water or oil) adheres to the sensor.
- Take countermeasures against extraneous light so that the light does not fall on the lenses of the sensor.

※ When highly accurate measurement is required, use shielding covers on the sensor.

If extraneous light enters the sensor when no object is present, use the timing input.

■ Power Supply

- All circuits in the sensor head are grounded to the case through a capacitor. However, noise conveyed through the power supply line may cause the sensor head to malfunction. Use a stabilized DC power supply having an insulated transformer.

  Be sure to ground the frame-grounding terminals and ground terminals when using switching regulators.

KEYENCE

Sensors for Factory Automation

KEYENCE

KEYENCE CORPORATION

2-13, Aketa-cho, Takatsuki, Osaka 569, Japan
TELEX: 5336530 KEYENC J
TEL: 81-726-84-2233 FAX: 81-726-84-0444

KEYENCE CORPORATION OF AMERICA
Head office:
17-17 Route 208 North, Fair Lawn, NJ 07410
TEL: 201-791-8811 800-328-2238 (all states)
FAX: 201-791-5791

California office:
3858 Carson Street, Suite 203, Torrance, CA 90503
TEL: 213-540-2254 800-543-8690 FAX: 213-316-1032

KEYENCE DEUTSCHLAND GmbH
Friedrich-List-Strass 44
7022 Leinfelden-Echterdingen,
West Germany
(Scheduled to be open in July 1990)

900601K Printed in Japan

เอกสารอ้างอิง

1. Jame T.Luxon ,David E.Parker ,Industrial laser and their applications ,Prentice-Hall International Editions,1992

2. Ernest O.Doebelin ,Measurement systems application and design, Fourth Edition ,PP 364-371 ,McGraw-Hill International Editions ,1990

3. The Engineering Staff of Analog Device ,Inc. in Analog-Digital Conversion Handbook Prentice-Hall ,Englewood Cliffs , NJO7632 ,1986

4. พิศาล งามเมืองแมน ,สราวุฒิ วณิชชานุกร " ระบบสแกนพื้นผิวโดยเลเซอร์ " วิทยานิพนธ์ระดับปริญญาตรี ภาควิชาฟิสิกส์ประยุกต์ คณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ปีการศึกษา 2535

5. วีรเทพ วรรณธนาเลิศ ,วีระศักดิ์ วิทูกิจวัฒนา ,วุฒิชัย ตันเปาว์ " การพัฒนาโต๊ะควบคุมด้วยระบบคอมพิวเตอร์สำหรับเครื่องคาร์บอนไดออกไซด์เลเซอร์ " วิทยานิพนธ์ระดับปริญญาตรี ภาควิชาฟิสิกส์ประยุกต์ คณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าธนบุรี ปีการศึกษา 2534

6. ดนัย อร่ามธรรมาพร ,นรินทร์ อติวงศ์แสงทอง ,อ.สมศักดิ์ เชียร์ศิริกุล , รศ.ดร.สมเกียรติ ศุภเดช " ตัวตรวจจับที่ไวต่อตำแหน่งแบบสองแกน " การประชุมวิชาการทางวิศวกรรมทางไฟฟ้าครั้งที่ 16 หน้า 374-379 คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

7. ยืน ภู่วรวรรณ ,เทคโนโลยีฮาร์ดแวร์ IBM PC ,ซีเอ็ดยูเคชั่น ,2533

8. ยืน ภู่วรวรรณ ,ทฤษฎีและการประยุกต์ไมโครโปรเซสเซอร์ Z-80 ,ซีเอ็ดยูเคชั่น ,2533

9. วัฒนศักดิ์ เทพกูล "เครื่องจักรกลดิจิตอล" คอมพิวเตอร์ อิเลคทรอนิคส์เวิร์ด 142(2536) : 75-84

10. นุกูล กระจาย ,การเขียนโปรแกรมและประมวลผลข้อมูลด้วยเทอร์โบปาสคาล, ซีเอ็ดยูเคชั่น ,2535

ประวัติผู้ทำโครงงาน

นายรุ่งเรือง แซ่โค้ว เกิดเมื่อวันที่ 8 พฤจิกายน พ.ศ. 2514 ที่จังหวัดกรุงเทพฯ สำเร็จการศึกษาระดับประถมศึกษาจากโรงเรียนแม่พระฟาติมาเมื่อปีการศึกษา 2527 ระดับมัธยมศึกษาตอนต้นจากโรงเรียนสามเสนวิทยาลัยเมื่อปีการศึกษา 2530 ระดับมัธยมศึกษาตอนปลาย (ม.5) จากโรงเรียนสามเสนวิทยาลัย เมื่อปีการศึกษา 2532 และเข้าศึกษาต่อในระดับอุดมศึกษาในสาขาวิชาฟิสิกส์ประยุกต์ คณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง สำเร็จการศึกษาในปีการศึกษา 2536 เลขประจำตัว 33504030

ประวัติการฝึกงาน

- สำนักงานพัฒนาเทคโนโลยี กลุ่มบริษัทพรีเมียร์ เมื่อ 21 มีนาคม - 7 มิถุนายน พ.ศ.2537 ผลงาน - ค้นคว้าและทำโครงงาน Electroflocculation

ประวัติผู้ทำโครงงาน

นายสมศักดิ์ ปัญทุวงศ์ เกิดเมื่อวันที่ 4 กุมภาพันธ์ พ.ศ. 2515 ที่จังหวัดนราธิวาส สำเร็จการศึกษาระดับประถมศึกษาจากโรงเรียนบ้านสุไหงโก-ลกเมื่อปีการศึกษา 2526 ระดับมัธยมศึกษาตอนต้นจากโรงเรียนสุไหงโก-ลกเมื่อปีการศึกษา 2529 ระดับมัธยมศึกษาตอนปลายจากโรงเรียนสวนกุหลาบวิทยาลัย เมื่อปีการศึกษา 2532 และเข้าศึกษาต่อในระดับอุดมศึกษาในสาขาวิชาฟิสิกส์ประยุกต์ คณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง สำเร็จการศึกษาในปีการศึกษา 2536 เลขประจำตัว 33504038

ประวัติการฝึกงาน

- สำนักงานพัฒนาเทคโนโลยี กลุ่มบริษัทพรีเมียร์ เมื่อ 21 มีนาคม - 7 มิถุนายน พ.ศ.2537 ผลงาน - ค้นคว้าและทำโครงงาน Ultrasonic Flaw Detector