เครื่องอ่านภาพแผ่นโลหะเรียบ

นาย ปรเมษฐ์ จินุพันธุ์

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต
ภาควิชาวิศวกรรมไฟฟ้า
บัณฑิตวิทยาลัย จุฬาลงกรณ์มหาวิทยาลัย
พ.ศ. 2537

ISBN 974-584-336-9

ลิขสิทธิ์ของบัณฑิตวิทยาลัย จุฬาลงกรณ์มหาวิทยาลัย

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

T174930666

FLAT METAL SHEET IMAGE SCANNER

Mr. PORAMASTE JINUPUN

A Thesis Submitted in Partial Fulfillment of the Requirements

for the Degree of Master of Engineering

Department of Electrical Engineering

Graduate School

Chulalongkorn University

1994

ISBN 974-584-336-9

หัวข้อวิทยานิพนธ์	เครื่องอ่านภาพแผ่นโลหะเรียบ
โดย	นาย ปรเมษฐ์ จินุพันธุ์
ภาควิชา	วิศวกรรมไฟฟ้า
อาจารย์ที่ปรึกษา	รศ. กฤษดา วิศวธีรานนท์

บัณฑิตวิทยาลัย จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้นับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญามหาบัณฑิต

_____ คณบดีบัณฑิตวิทยาลัย
(ศาตราจารย์ ดร. ถาวร วัชราภัย)

_____ ประธานกรรมการ
(ดร. สมบูรณ์ จงชัยกิจ)

_____ อาจารย์ที่ปรึกษา
(รองศาตราจารย์ กฤษดา วิศวธีรานนท์)

_____ กรรมการ
(รองศาตราจารย์ ดร. เอกชัย ลีลารัศมี)

_____ กรรมการ
(นาย สินชัย เจียมดำรัศ)

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

พิมพ์ต้นฉบับบทคัดย่อวิทยานิพนธ์ภายในกรอบสีเขียวนี้เพียงแผ่นเดียว

ประเมษฐ์ จินุพันธุ์ : เครื่องอ่านภาพแผ่นโลหะเรียบ
(Flat Metal Sheet Image Scanner)
อาจารย์ที่ปรึกษา : รศ. กฤษดา วิศวธีรานนท์
123 หน้า. ISBN 974-584-336-9

วิทยานิพนธ์นี้กล่าวถึงการออกแบบและการสร้างเครื่องอ่านภาพแผ่นโลหะเรียบ เพื่อใช้ตรวจวัดตำแหน่งของรูเจาะบนแผ่นโลหะ โดยใช้งานร่วมกับโปรแกรมวิเคราะห์ความคลาดเคลื่อนของตำแหน่งรูเจาะบนแผ่นโลหะ เครื่องที่สร้างขึ้นนี้ สามารถอ่านภาพแผ่นโลหะขนาดสูงสุดไม่เกิน 1 X 2.4 X 0.03 เมตร โดยใช้ตัวอ่านภาพเชิงเส้นชนิด CCD อ่านกวาดภาพตามหน้ากว้าง 1 เมตร และ ใช้ชุดพัลส์มอเตอร์ขับเกลียวตัวหนอนเคลื่อนหัวอ่านไปตามแนวความยาว 2.4 เมตร การอ่านภาพตามหน้ากว้าง 1 เมตร นั้น ใช้จุดรับภาพ 20,000 จุด โดยใช้กล้อง 5 ตัว เทคนิคในการลดทอนข้อมูลภาพใช้วิธีการเก็บเฉพาะตำแหน่งที่เป็นขอบของรูในแผ่นโลหะเท่านั้น การปรับแต่ง และ การต่อภาพที่ได้จากกล้องแต่ละตัวทำด้วยซอฟต์แวร์ เครื่องต้นแบบที่พัฒนาขึ้นนี้ให้ความแม่นยำในการอ่านภาพประมาณ 0.6 มิลลิเมตร และ ให้ความละเอียดของภาพ 508 จุดต่อนิ้ว การอ่านภาพเต็มระยะพิกัดใช้เวลาประมาณ 20 นาที

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ภาควิชา วิศวกรรมไฟฟ้า ลายมือชื่อนิสิต ประเมษฐ์ จินุพันธุ์
สาขาวิชา วิศวกรรมไฟฟ้า ลายมือชื่ออาจารย์ที่ปรึกษา
ปีการศึกษา 2536 ลายมือชื่ออาจารย์ที่ปรึกษาร่วม

C415518 : MAJOR ELECTRICAL ENGINEERING.
KEY WORD. IMAGE SCANNER / CCD IMAGE SENSOR / METAL SHEET INSPECTION.

PORAMASTE JINUPUN : FLAT METAL SHEET IMAGE SCANNER.

THESIS ADVISOR : ASSOC.PROF.KRISADA VISAVATEERANON.

123 pp. ISBN 974-584-336-9

This thesis presents design and construction of a flat metal sheet image scanner, which is a major tool for automatic inspection of a punched metal sheet by using microcomputer. It can read the image of a metal sheet up to the maximum dimension of 1 X 2.4 X 0.03 meter. This machine utilizes linear image sensors for scanning image data along the width and uses a pulse motor driving a worm gear for moving the head assembly along the length of the metal sheet. By using 5 cameras, one scan line data can be composed of 20,000 pixels. The data compression technique in reducing the image data is the extraction of the edge position of holes along a scan line. Image data from 5 cameras are combined, compensated and calibrated by using a designed calibration program. The developed prototype has a maximum absolute accuracy of 0.6 mm and 508 dots per inch resolution. And the scanning time for a full size metal sheet is about 20 minutes.

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ภาควิชา วิศวกรรมไฟฟ้า ลายมือชื่อนิสิต ปรเมษฐ์ จินุพันธุ์

สาขาวิชา วิศวกรรมไฟฟ้า ลายมือชื่ออาจารย์ที่ปรึกษา

ปีการศึกษา 2536 ลายมือชื่ออาจารย์ที่ปรึกษาร่วม

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้ ได้สำเร็จลุล่วงไปได้ด้วยดี ด้วยความช่วยเหลืออย่างดียิ่งของ รองศาสตราจารย์ กฤษดา วิศวธีรานนท์ อาจารย์ที่ปรึกษาวิทยานิพนธ์ ซึ่งท่านได้กรุณาให้คำแนะนำ และข้อคิดเห็นต่างๆของการวิจัยด้วยดีตลอด จึงขอขอบพระคุณท่าน ณ ที่นี้ และขอขอบคุณ คุณ สินชัย เจียมดำรัส ในนามของ บริษัท ลิงค์คาร์ไลน์ จำกัด ที่ได้ให้ความอนุเคราะห์ ประสานงานที่ดี ทั้งทางด้านทุนการวิจัย การจัดซื้ออุปกรณ์ต่างๆ รวมทั้งการทำชุดทดสอบการทำงานให้

ท้ายนี้ผู้วิจัยใคร่ขอขอบคุณ คุณ ศวรรยา แป้นเหมือน ที่ช่วยในการพิมพ์และจัดแต่งวิทยานิพนธ์ และเป็นกำลังใจแก่ผู้วิจัยเสมอมาจนสำเร็จการศึกษา

และขอขอบคุณพระเจ้า และ พ่อแม่ที่แท้จริง ที่ได้ประทานสติปัญญาที่ดี และ ให้การเปิดเผยถึงหนทางอันนำมาซึ่งความสำเร็จนี้แก่ข้าพเจ้า ข้าพเจ้าขอสดุดี ณ ที่นี้

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

สารบัญ

หน้า

บทคัดย่อภาษาไทย..... ง

บทคัดย่อภาษาอังกฤษ..... จ

กิตติกรรมประกาศ..... ฉ

สารบัญตาราง..... ฌ

สารบัญภาพ..... ฎ

บทที่ 1 บทนำ..... 1

1.1 ความเป็นมา..... 1

1.2 ขั้นตอนการออกแบบระบบ..... 2

1.3 วัตถุประสงค์ของการวิจัย..... 4

1.4 ขอบเขตของการวิจัย..... 5

1.5 วิธีวิจัยโดยย่อ..... 5

1.6 ประโยชน์ที่คาดว่าจะได้รับ..... 5

บทที่ 2 ทฤษฎี และ หลักการทำงานของอิมเมจสแกนเนอร์..... 6

2.1 อิมเมจสแกนเนอร์ คืออะไร..... 6

2.2 ลักษณะของอิมเมจสแกนเนอร์แบบต่างๆ..... 7

2.2.1 ดรัมอิมเมจสแกนเนอร์..... 7

2.2.2 เทเบิลอิมเมจสแกนเนอร์ (ใช้เซ็นเซอร์ชนิดคอนแท็ค)..... 8

2.2.3 เทเบิลอิมเมจสแกนเนอร์ (ใช้เซ็นเซอร์ชนิดใช้กับเลนส์)..... 9

2.3 การติดต่อกับผู้ใช้..... 10

2.3.1 การเชื่อมต่อทางฮาร์ดแวร์ระหว่างเครื่องอิมเมจสแกนเนอร์กับเครื่องไมโครคอมพิวเตอร์... 10

สารบัญ(ต่อ)

หน้า

2.4 ลักษณะและการทำงานของอิมเมจเซ็นเซอร์เซ็นเซอร์.... 11
2.4.1 การทำงานของตัวอิมเมจเซ็นเซอร์..... 12
2.5 องค์ประกอบของการอ่านภาพ..... 14
2.5.1 เลนส์..... 14
2.5.2 ทิศทางของแสงและผลที่ได้..... 15
บทที่ 3 แนวความคิดในการออกแบบ..... 16
3.1 ข้อกำหนดของเครื่องอ่านภาพแผ่นโลหะเรียบ..... 16
3.2 ส่วนประกอบฮาร์ดแวร์ของเครื่องอ่านภาพ
แผ่นโลหะเรียบ..... 16
3.3 เทคนิคการลดจำนวนข้อมูลจากการสแกนภาพ..... 19
3.4 การควบคุมการอ่านสัญญาณภาพจากอิมเมจเซ็นเซอร์..... 20
3.5 บล็อคไดอะแกรมการอ่านภาพ..... 21
3.6 การทดลองเพื่อหาวิธีการที่เหมาะสม..... 24
3.6.1 วงจรต่าง ๆ ของการทดลอง..... 24
3.6.2 ผลการทดลอง และปัญหาที่พบ..... 27
3.6.3 การแก้ไขปัญหา..... 30
บทที่ 4 การสร้างเครื่องอ่านภาพแผ่นโลหะเรียบ..... 34
4.1 การออกแบบวงจร..... 34
4.1.1 บอร์ดไมโครคอนโทรลเลอร์..... 34
4.1.2 เมนบอร์ด..... 35
4.1.3 การ์ดควบคุมกล้อง..... 37
4.1.4 การ์ดวงจรการสื่อสารข้อมูล..... 37
4.1.5 บอร์ดวงจรขยายสัญญาณภาพ..... 38
4.1.6 บอร์ดอินเตอร์เฟส..... 38

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

สารบัญ(ต่อ)

หน้า

4.1.7 สเต็ปปิ้งมอเตอร์คอนโทรลเลอร์
และการต่อใช้งาน..... 39
4.2 การออกแบบโปรแกรม..... 44
4.2.1 โปรแกรมการสื่อสารเพื่อเชื่อมต่อการทำงาน
ระหว่างเครื่องอ่านภาพแผ่นโลหะเรียบ
กับเครื่องไมโครคอมพิวเตอร์..... 46
4.2.1.1 โปรแกรมการรับ-ส่งคำสั่ง..... 47
4.2.1.2 โปรแกรมการรับ-ส่งข้อมูล..... 48
4.2.2 โปรแกรมในส่วนของเครื่องไมโครคอมพิวเตอร์. 49
4.2.2.1 โครงสร้างของโปรแกรมในเครื่อง
ไมโครคอมพิวเตอร์..... 49
4.2.2.2 โปรแกรมแสดงผลกราฟฟิก..... 50
4.2.3 โปรแกรมในส่วนของเครื่องอ่านภาพ
แผ่นโลหะเรียบ..... 52
4.2.3.1 การแบ่งเวลาของโปรแกรม..... 52
4.2.3.2 โปรแกรมอินเตอร์รัพต์..... 52
4.2.3.3 โครงสร้างโปรแกรมหลักในเครื่อง
อ่านภาพแผ่นโลหะเรียบ..... 53
4.2.3.4 โปรแกรมการสแกน..... 53
4.2.3.5 โปรแกรมเปลี่ยนค่าเทรสโฮลด์.... 55
4.2.3.6 โปรแกรมการหาจุดเริ่มต้น
ของการสแกน..... 55
4.2.4 โปรแกรมรวมภาพและแก้ไขภาพ..... 56
4.2.4.1 ระบบไฟล์ที่ใช้..... 56

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

สารบัญ(ต่อ)

หน้า

4.2.4.2 การต่อภาพและการแก้ไขความผิดเพี้ยนจากการสแกน..... 56
4.2.4.4 โปรแกรมการต่อภาพ และการแก้ไขภาพ..... 57
บทที่ 5 การปรับเทียบ การทดสอบ และการวิเคราะห์ผล..... 60
5.1 การปรับเทียบ..... 60
5.2 การทดสอบการทำงานของเครื่องอ่านภาพแผ่นโลหะเรียบ. 62
5.2.1 ความเร็วการเคลื่อนที่ของหัวอ่าน..... 62
5.2.2 ผลของความเร็วต่อจำนวนช่องในการการสแกน. 63
5.2.3 ขนาดความถูกต้องที่ได้จากการสแกน..... 64
5.3 การทดสอบโปรแกรมแก้ไขภาพและการรวมภาพ..... 70
5.3.1 ผลของความเร็วต่อจำนวนช่องในการแก้ไขภาพและรวมภาพ..... 70
5.3.2 ปริมาณของข้อมูลต่อจำนวนของช่อง..... 71
5.4 การวิเคราะห์ผล..... 72
บทที่ 6 สรุปและข้อเสนอแนะ..... 77
6.1 สรุป..... 77
6.2 ปัญหาและข้อเสนอแนะ..... 78
เอกสารอ้างอิง..... 81
ภาคผนวก..... 82
ภาคผนวก ก. วงจรอิเล็กทรอนิกส์..... 83
ภาคผนวก ข. โปรแกรม..... 88
ประวัติผู้เขียน.....123

บทที่ 1

บทนำ

1.1 ความเป็นมา

ในปัจจุบันนี้จะเห็นได้ว่า โรงงานอุตสาหกรรมการผลิตกำลังมีการแข่งขันกันอย่างมาก ดังนั้นจึงได้มีการพัฒนาระบบการผลิตให้มีประสิทธิภาพสูง และที่จะกล่าวถึงในที่นี้ คือ อุตสาหกรรมที่มีการตัด,การเจาะแผ่นโลหะ เช่น โรงงานทำ แอร์ ตู้คอนโซล ตู้เอกสาร เฟอร์นิเจอร์ เป็นต้น ซึ่งการควบคุมผลผลิตให้มีคุณภาพที่ดีนั้น จะต้องมีการตรวจวัดผลผลิตว่ามีคุณภาพตามที่ต้องการหรือไม่ และมีความคลาดเคลื่อนเพียงใด เพื่อนำมาใช้ในการปรับแต่งเครื่องจักรให้ผลิตผลงานออกมามีคุณภาพตามที่ต้องการ ซึ่งเดิมนั้นใช้คนมาทำการตรวจวัดโดยวัดด้วยมือซึ่งล้าช้า และเกิดการผิดพลาดได้ง่าย ดังนั้นจึงได้นำวิทยาการทางด้านอิเล็กทรอนิกส์คอมพิวเตอร์มาสร้างระบบตรวจวัด ซึ่งให้ความผิดพลาดน้อย และ รวดเร็วกว่า

การพัฒนาโดยการนำระบบคอมพิวเตอร์มาช่วยในการตรวจวัด คุณภาพนั้นสามารถนำข้อมูลที่ได้มาเข้าขบวนการต่าง ๆ ตามต้องการ ซึ่งก็มีซอฟต์แวร์มากมายที่สามารถนำข้อมูลมาช่วยในการวิเคราะห์ผลผลิต และยังสามารถจัดทำเอกสารเพื่อรายงานผลการวิเคราะห์ได้สะดวก ดังนั้นจึงทำให้การจัดระบบงานในอุตสาหกรรมมีประสิทธิภาพมากยิ่งขึ้น

การตรวจวัดโดยใช้ระบบคอมพิวเตอร์มาช่วยนี้ มีโครงสร้างซึ่งแบ่งเป็น 2 ส่วนคือ

1) ส่วนการอ่านภาพแผ่นโลหะ

ส่วนนี้จะทำหน้าที่ในการอ่านภาพเพื่อเก็บขนาด และรูปร่างของแผ่นโลหะมาเก็บไว้ในระบบสำรองข้อมูล

2) ส่วนของซอฟต์แวร์ที่ใช้วิเคราะห์ข้อมูล

ส่วนนี้จะทำหน้าที่นำข้อมูลที่ได้จากการอ่านภาพแผ่นโลหะ มาทำการเก็บไว้ในระบบสำรองข้อมูล มาเข้ากระบวนการวิเคราะห์และเปรียบเทียบ

ผลที่ได้เพื่อหาความคลาดเคลื่อนที่เกิดขึ้น

จากโครงสร้างของระบบที่มีขนาดใหญ่ จึงได้แยกออกมาทำการพัฒนาเฉพาะส่วนการอ่านภาพของแผ่นชิ้นงาน ซึ่งการเชื่อมต่อกันของ 2 ส่วนนี้ ทำได้โดยโครงสร้างข้อมูลที่มีมาตรฐานเดียวกัน ซึ่งจากนี้ไปจะขอกล่าวถึงเฉพาะส่วนการอ่านภาพแผ่นโลหะเท่านั้น

ส่วนการอ่านภาพนี้เรียกรวมๆว่า "อิมเมจสแกนเนอร์"(Image Scanner) โดยเราจำเป็นต้องออกแบบสร้างขึ้นเอง เนื่องจากยังไม่มีอิมเมจสแกนเนอร์แบบที่ใช้เฉพาะกับแผ่นโลหะ ซึ่งมีความหนาและมีขนาดที่ใหญ่เกินกว่าแบบที่มีขายโดยทั่วไปจะทำงานได้ (ให้ดูรายละเอียดในบทที่ 2) เครื่องอิมเมจสแกนเนอร์ที่ออกแบบมาใช้กับแผ่นโลหะ ในวิทยานิพนธ์เล่มนี้ตั้งชื่อเรียกว่า"เครื่องอ่านภาพแผ่นโลหะเรียบ"(Flat Metal Sheet Image Scanner) โดยสามารถอ่านภาพแผ่นโลหะครอบคลุมเนื้อที่ขนาด กว้าง 1 เมตร ยาว 2.4 เมตร และมีความแม่นยำของตำแหน่งที่ต้องการ คือ 0.2 มิลลิเมตร

1.2 ขั้นตอนการออกแบบระบบ

การออกแบบระบบนั้นมีขั้นตอนดังแสดงในรูป 1.1 [1][2]

1) ให้ข้อกำหนดรายละเอียดของระบบ เป็นขั้นแรกของการออกแบบระบบทั่วไป คือ การให้ข้อกำหนดรายละเอียดของระบบ ข้อกำหนดนี้จะเป็นตัวบอกว่าระบบนั้นจะต้องทำงานอะไรได้บ้าง แต่ยังไม่ครอบคลุมถึงวิธีการที่ใช้ในการทำงานนั้น

2) การออกแบบระบบเป็นระดับบล็อก เป็นการกำหนด อินพุต,เอาต์พุต และแบ่งหน้าที่การทำงานของส่วนต่างๆ โดยจะนำการออกแบบในระดับบอร์ดมาใช้ ซึ่งทำให้ลดระยะเวลา และง่ายต่อการออกแบบ ในช่วงนี้ก็ได้แบ่งส่วนที่เป็นฮาร์ดแวร์และซอฟต์แวร์ ซึ่งทำให้สามารถทำการออกแบบทั้งสองส่วนขนานกันไปได้

3) การออกแบบฮาร์ดแวร์ โดยจะทำการออกแบบและเลือกอุปกรณ์ที่จะใช้ในวงจร รวมทั้งการการต่อวงจรทั้งหมดเป็นระบบ

4) การสร้างเครื่องต้นแบบ การสร้างได้นำการออกแบบระดับบอร์ดมาใช้ ดังนั้นบางส่วนของระบบที่สามารถหาซื้อได้ง่ายก็จะไม่ทำการสร้างแต่จะ

นำมาใช้ส่วนที่ทำก็จะเป็นส่วนของการอินเตอร์เฟส และ อินพุตของระบบ เท่านั้น

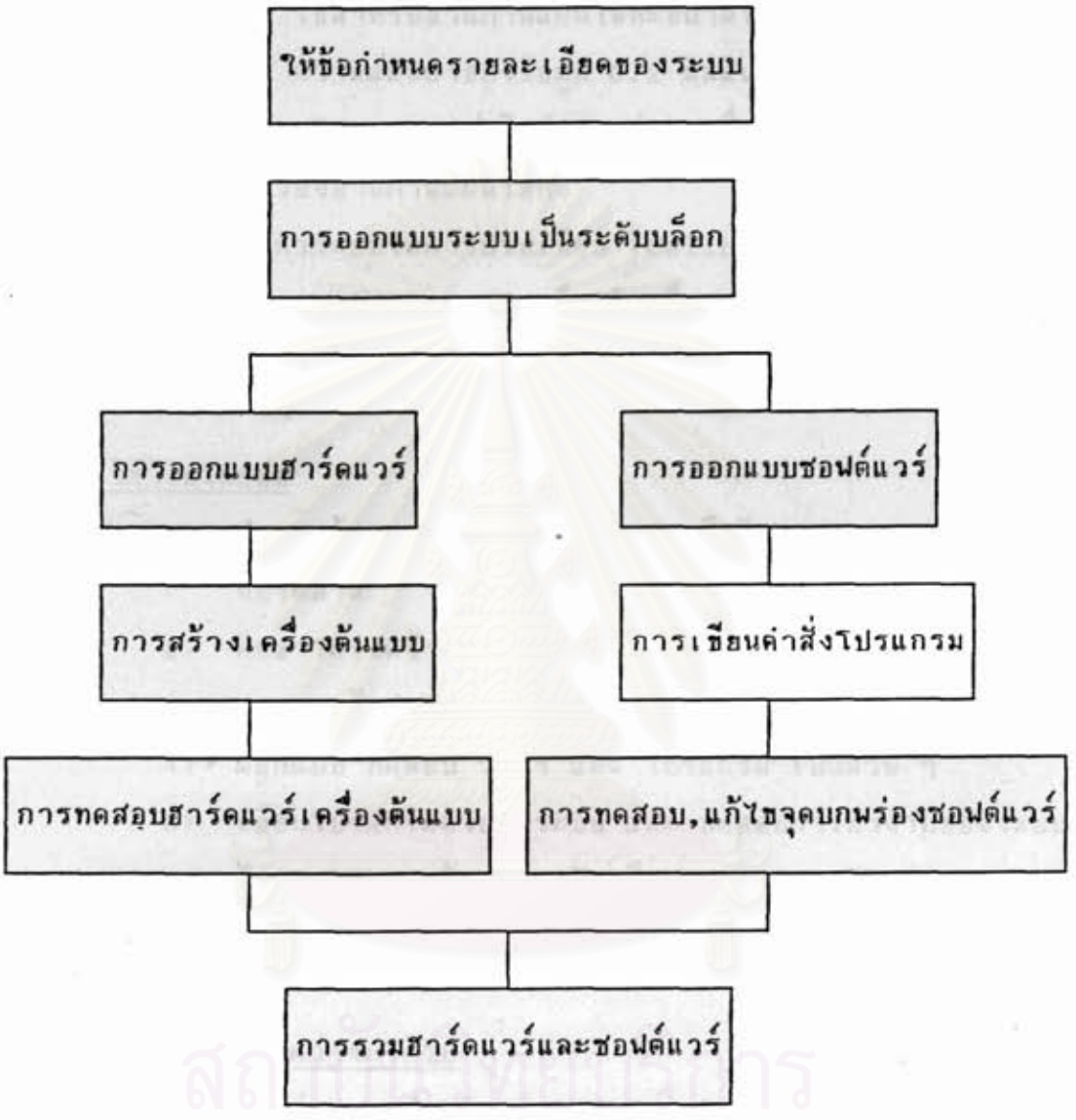
5) การทดสอบฮาร์ดแวร์ของเครื่องต้นแบบ การทดสอบทำขึ้นเพื่อตรวจสอบความถูกต้อง ของการทำงานของระบบและวงจร โดยใช้โปรแกรมสั้น ๆ และเมื่อเจอกับข้อบกพร่องก็จะทำการแก้ไข

6) การออกแบบซอฟต์แวร์ เริ่มโดยการแบ่งโปรแกรมออกเป็นโมดูลจากลักษณะการทำงาน และ จัดลำดับการทำงานของแต่ละส่วนของโปรแกรม จากนั้นก็กระจายออกเป็นส่วนย่อยตามหน้าที่ เพื่อนำมาเขียนเป็นโปรแกรมจริงต่อไป

7) การเขียนคำสั่งโปรแกรม การเขียนโปรแกรมโดยส่วนใหญ่พยายามใช้ภาษาระดับสูง เพื่อความสะดวกต่อการพัฒนา แต่ในบางส่วนที่ต้องการความเร็วการทำงานสูงก็จะเขียนด้วยภาษาระดับต่ำ และมีการแบ่งในลักษณะเป็นโมดูล

8) การทดสอบและแก้ไขจุดบกพร่องของซอฟต์แวร์ การทดสอบส่วนของซอฟต์แวร์นี้ ทำได้โดยการตรวจสอบสถานะต่าง ในระหว่างรันโปรแกรม ซึ่งก็มีทั้งแบบเป็นขั้น(step test)เพื่อตรวจสอบลำดับการทำงาน และแบบเวลาจริง (real time test) เพื่อตรวจสอบการทำงานจริงว่าสามารถให้ข้อมูลที่ถูกต้อง ถ้ามีจุดบกพร่องก็จะได้ทำการแก้ไขต่อไป กรณีของซอฟต์แวร์ในส่วนของไมโครคอนโทรลเลอร์นั้น ในช่วงทดสอบจะใช้ ROM EMULATOR ช่วยในการ Down Load โปรแกรม จนกระทั่งโปรแกรมสมบูรณ์แล้วจึงนำไปโปรแกรมลง EPROM จริงๆ

9) การรวมฮาร์ดแวร์และซอฟต์แวร์ เพื่อตรวจสอบความถูกต้องของระบบโดยรวมทั้งหมด ในส่วนของการประสานการทำงานระหว่างฮาร์ดแวร์และ ซอฟต์แวร์ซึ่งเป็นผลทดสอบท้ายสุดก่อนที่จะตัดสินความถูกต้อง และเมื่อผ่านขั้นตอนนี้ไปการพัฒนาก็จะเสร็จสิ้น



รูปที่ 1.1 ขั้นตอนการออกแบบระบบ

1.3 <u>วัตถุประสงค์ของการวิจัย</u>

ออกแบบสร้างเครื่องอ่านภาพสำหรับแผ่นโลหะขนาดใหญ่ ซึ่งทำงานร่วมกับเครื่องคอมพิวเตอร์ โดยเน้นถึงความแม่นยำของข้อมูลและความเร็วในการอ่านข้อมูลภาพ รวมทั้งสามารถนำข้อมูลที่ได้จากเครื่องอ่านภาพไปใช้กับซอฟต์แวร์อื่น ๆ เพื่อที่จะได้ทำการวิเคราะห์ผลต่อไป

1.4 ขอบเขตของการวิจัย

1) ออกแบบสร้างเครื่องอ่านภาพแผ่นโลหะเรียบ ที่มีข้อกำหนดดังนี้
   - ใช้สำหรับอ่านภาพแผ่นโลหะขนาดไม่เกิน 1เมตรX2.4เมตร
   - ความแม่นยำของข้อมูล 0.2 มิลลิเมตร
2) พัฒนาโปรแกรมการติดต่อระหว่างเครื่องไมโครคอมพิวเตอร์ กับเครื่องอ่านภาพแผ่นโลหะ
3) พัฒนาระบบในการปรับเทียบ (Calibrate) และระบบการปรับแต่ง (Alignment) เพื่อให้เครื่องภาพให้ข้อมูลออกมาถูกต้อง
4) ทดสอบการทำงานของเครื่องอ่านภาพที่สร้างขึ้น

1.5 วิธีวิจัยโดยย่อ

1) สำรวจข้อมูลของเครื่องอ่านภาพ หรืออิมเมจสแกนเนอร์ที่มีอยู่ในท้องตลาด
2) ศึกษาโครงสร้างและการทำงานของเครื่องอ่านภาพจากเอกสาร
3) แจกแจงข้อกำหนด และ ออกแบบ ระบบการอ่านภาพ
4) ออกแบบ ทดสอบ วงจร และ โปรแกรม เป็นส่วน ๆ
5) เขียนโปรแกรมควบคุมระบบ และ ทดสอบการทำงานของระบบ
6) พัฒนาระบบทางด้านการปรับเทียบ และ การปรับแต่ง
7) เขียนวิทยานิพนธ์

1.6 ประโยชน์ที่คาดว่าจะได้รับ

ได้เครื่องอ่านภาพสำหรับแผ่นโลหะขนาดใหญ่ ที่สามารถนำไปพัฒนาใช้งานได้จริง เช่น ใช้ในการอ่านภาพแผ่นโลหะที่มีการเจาะและบั้มในโรงงานอุตสาหกรรมนำมาตรวจสอบความถูกต้องของการเจาะและบั้มแผ่นโลหะ โดยมีราคาถูกกว่าที่ซื้อจากต่างประเทศ และได้ความรู้ที่จะเป็นแนวทางในการออกแบบเครื่องอ่านภาพแบบอื่นๆ นอกจากนี้ยังเป็นการพัฒนาเทคโนโลยีทางด้านการอ่านภาพให้มีประสิทธิภาพมากยิ่งขึ้น

บทที่ 2

ทฤษฎี และ หลักการทำงานของเครื่องอิมเมจสแกนเนอร์

2.1 อิมเมจสแกนเนอร์ คืออะไร

อิมเมจสแกนเนอร์ คือเครื่องมือที่ใช้สำหรับอ่านภาพ โดยทำการเปลี่ยนรูปแบบสัญญาณภาพให้เป็นสัญญาณดิจิตอลด้วยตัวเซ็นเซอร์ ซึ่งส่วนใหญ่ตัวเซนเซอร์ที่ใช้มักจะเป็นตัวอ่านภาพเชิงเส้น(Linear Image Sensor) ดังนั้นการอ่านจะต้องมีการเลื่อนตำแหน่งการอ่านด้วยจึงจะได้ภาพทั้งหมด

ลักษณะและองค์ประกอบของเครื่องอิมเมจสแกนเนอร์มักคล้ายๆกัน คือ

1) ส่วนที่เก็บข้อมูลจากภาพ
2) ส่วนการเคลื่อนไหวเพื่อเลื่อนตำแหน่งการอ่านภาพ
3) ส่วนที่ให้แสงสว่างแก่ภาพ
4) ส่วนที่เชื่อมต่อกับระบบคอมพิวเตอร์

ส่วนที่ใช้สำหรับรับภาพนั้น มักนิยมใช้อิมเมจเซ็นเซอร์ที่ชนิดเป็น CCD (Charge Coupled Devices) และมีวงจรควบคุม ทั้งส่วนการอ่านภาพ และการควบคุมระบบ โดยจะยังไม่ขอกล่าวในรายละเอียด

ส่วนการเคลื่อนไหวเพื่อเลื่อนตำแหน่งการอ่านภาพ ถ้าเป็นขนาดมือถือก็จะใช้มือในการเลื่อนตำแหน่งการอ่าน และถ้าเป็นขนาดใหญ่ก็จะมอเตอร์ในการขับเคลื่อน ซึ่งก็มี 2 ลักษณะ คือ เลื่อนตัวอ่านภาพ(แบบโต๊ะและแบบมือถือ) หรือ เลื่อนภาพที่จะอ่าน(แบบดรัมอิมเมจสแกนเนอร์)

ส่วนการให้แสงสว่างนั้น ถ้าเป็นขนาดเล็กจะใช้ LED หรือ หลอดไฟเล็กๆ และถ้าเป็นขนาดใหญ่ขึ้นมาจะใช้หลอดที่มีไส้ยาว หรือ หลอดที่มีลักษณะยาว เช่น หลอดฟลูออเรสเซ็นต์ หลอดนีออน เป็นต้น ฯลฯ โดยแสงที่ใช้จะต้องสม่ำเสมอและไม่กระพริบ หรือ กระเพื่อม

ส่วนที่เชื่อมต่อกับระบบคอมพิวเตอร์ เพื่อนำข้อมูลจากการสแกนส่งเข้าสู่ระบบคอมพิวเตอร์ โดยมีทางเข้าของข้อมูลหลายทาง เช่น พอร์ตอนุกรม พอร์ตขนาน และ SCSI การเลือกใช้ต้องเลือกตามความเหมาะสมกับจำนวนข้อ

มูล ซึ่งก็ขึ้นกับพื้นที่ของการสแกน ความละเอียดที่ต้องการ และความเร็วของการส่งข้อมูลของแบบนั้นๆ

ในปัจจุบันนี้ได้มีอิมเมจสแกนเนอร์ออกจำหน่ายมากมาย แต่ก็ไม่สามารถนำมาใช้กับการอ่านภาพแผ่นโลหะ และส่วนใหญ่มักจะมีขนาดเล็ก อย่างไรก็ตามเราสามารถนำทฤษฎีและหลักการมาใช้กันได้

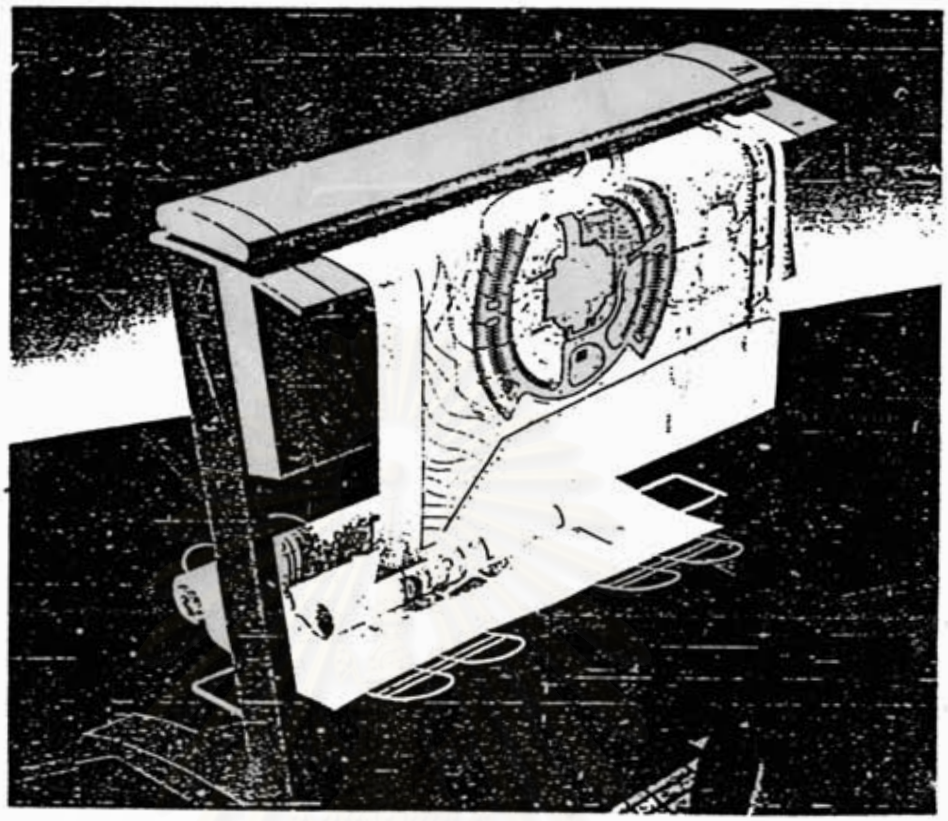
2.2 ลักษณะของอิมเมจสแกนเนอร์แบบต่าง ๆ

โดยจะได้อธิบายลักษณะเด่น รวมทั้งข้อดีและข้อเสียของแต่ละแบบ ดังต่อไปนี้

2.2.1 ดรัมอิมเมจสแกนเนอร์(Drum Image Scanner) ลักษณะเด่นของแบบนี้ คือ จะมีระบบการเลื่อนแผ่นกระดาษโดยลูกกลิ้ง และตัวอ่านภาพเชิงเส้นจะตรึงอยู่กับที่ ขณะที่กระดาษเคลื่อนที่ผ่านหัวอ่าน ดังรูปที่ 2.1 [3]

เครื่องแบบนี้มักสร้างให้สามารถใช้กับกระดาษแผ่นใหญ่ ในรูปที่ 2.1 สามารถใช้อ่านภาพจากกระดาษได้ถึงขนาด A0 ดังนั้นจึงต้องใช้ตัวอิมเมจเซ็นเซอร์หลายตัวและมักจะเป็นชนิดที่ใช้กับเลนส์ โดยแสงสว่างที่ใช้กับเครื่องจะใช้หลอดฟลูออเรสเซ็นต์ ในการอ่านภาพจะอาศัยการสะท้อนแสงจากภาพ เนื่องจากการเลื่อนตำแหน่งการอ่านภาพใช้ดรัมในการเลื่อนกระดาษจึงทำให้มีขนาดที่เล็กลง แต่ก็มีความคลาดเคลื่อนจากการลื่นไถลในขณะเลื่อนกระดาษเกิดขึ้น

ข้อดี	ข้อเสีย
1) ระบบการป้อนกระดาษทำได้ง่าย	1) ใช้กับแผ่นโลหะไม่ได้ เนื่องจากมีน้ำหนักมาก และมีความหนา
2) ส่วนกลไกใช้เนื้อที่น้อย	2) มีความคลาดเคลื่อนสูง
3) สามารถสร้างให้มีขนาดใหญ่ได้	

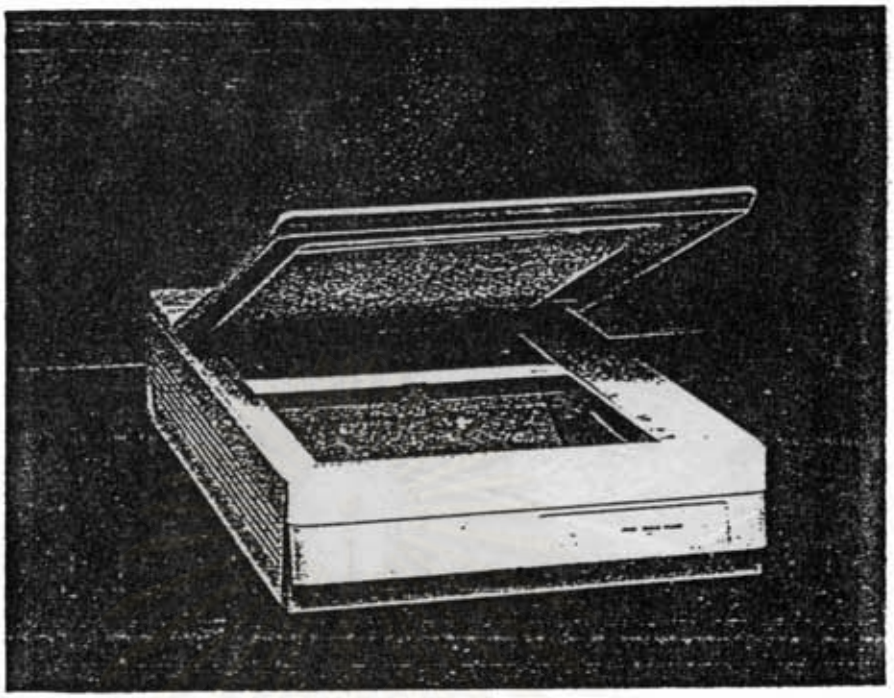


รูปที่ 2.1 ดรัมอิมเมจสแกนเนอร์

2.2.2 เทเบิลอิมเมจสแกนเนอร์(Table Image Scanner) (ใช้เซ็นเซอร์ชนิดคอนแท็ค) ลักษณะเด่นของอิมเมจสแกนเนอร์แบบนี้จะไม่มีการเลื่อนแผ่นกระดาษ แต่จะเลื่อนหัวอ่านในเวลาอ่านภาพ และตัวเซ็นเซอร์ต้องวางชิดติดกับภาพเป้าหมาย ดังรูปที่ 2.2 [4]

เครื่องแบบนี้มีลักษณะคล้ายกับเครื่องถ่ายเอกสารมาก มักมีขนาดไม่ใหญ่นัก ดังนั้นตัวอ่านภาพเชิงเส้นที่ใช้โดยส่วนใหญ่จึงเป็นชนิดคอนแท็ค (Contact Type Linear Image Sensor) โดยจะใช้แสงสว่างจากหลอดฮาโลเจน ในการอ่านภาพจะอาศัยการสะท้อนแสงจากภาพ การเลื่อนตำแหน่งการอ่านภาพจะแตกต่างจากแบบแรกโดยเลื่อนตัวอิมเมจเซ็นเซอร์แทน ดังนั้นส่วนกลไกการเคลื่อนไหวจึงซับซ้อนกว่าแบบแรก

ข้อดี	ข้อเสีย
1) สามารถใช้กับแผ่นโลหะได้	1) การทำส่วนเคลื่อนไหวยาก
2) มีความผิดพลาดน้อยกว่าแบบดรัมอิมเมจสแกนเนอร์	2) ส่วนกลไกใช้เนื้อที่มาก
	3) ไม่มีขนาดใหญ่ขาย

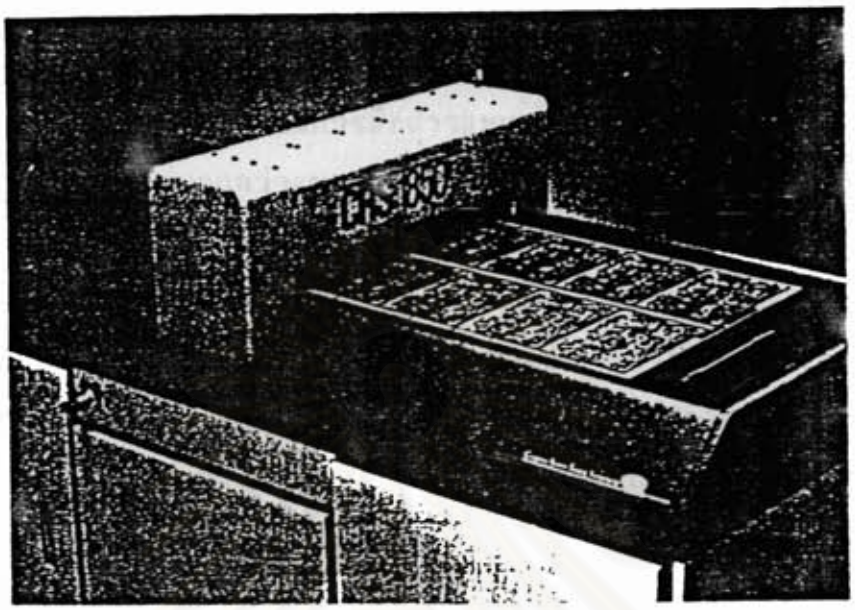


รูปที่ 2.2 เทเบิลอิมเมจสแกนเนอร์(ใช้เซ็นเซอร์ชนิดคอนแท็ค)

2.2.3 เทเบิลอิมเมจสแกนเนอร์(Table Image Scanner) (ใช้เซ็นเซอร์ชนิดใช้กับเลนส์) ลักษณะเด่นของอิมเมจสแกนเนอร์แบบนี้จะไม่มีการเลื่อนแผ่นกระดาษ แต่จะเลื่อนหัวอ่านในเวลาอ่านภาพ และจะต้องใช้เลนส์ในการลดขนาดของภาพเป้าหมายลงบนเซ็นเซอร์ ดังรูปที่ 2.3 [5]

เครื่องแบบนี้มีลักษณะคล้ายกับเครื่องแบบที่สอง ต่างกันตรงที่แบบที่สองตัวอิมเมจเซ็นเซอร์จะอยู่ข้างล่าง ทำให้ต้องใช้กระจกเป็นผิวของโต๊ะ ส่วนแบบนี้ตัวอยู่ข้างบนพื้นโต๊ะเป็นสีดำทึบ มักมีขนาดใหญ่ ดังนั้นจึงต้องใช้ตัวอิมเมจเซ็นเซอร์หลายตัวและมักจะเป็นชนิดที่ใช้กับเลนส์ โดยจะใช้แสงสว่างจากหลอดฟลูออเรสเซ็นต์ ในการอ่านภาพจะอาศัยการสะท้อนแสงจากภาพ การเลื่อนตำแหน่งการอ่านภาพจะแตกต่างจากแบบแรก โดยเลื่อนตัวอิมเมจเซ็นเซอร์แทน ดังนั้นส่วนกลไกการเคลื่อนไหวจึงซับซ้อนกว่าแบบแรก

ข้อดี	ข้อเสีย
1) สามารถใช้กับแผ่นโลหะได้	1) การทำส่วนเคลื่อนไหวยาก
2) ความคลาดเคลื่อนน้อยกว่าแบบดรัมอิมเมจสแกนเนอร์	2) ส่วนกลไกใช้เนื้อที่มาก
3) สร้างขนาดใหญ่ได้ง่าย	3) ยากต่อการปรับแต่ง



รูปที่ 2.3 เทเบิลอิมเมจสแกนเนอร์(ใช้เซ็นเซอร์ชนิดใช้กับเลนส์)

จากลักษณะของอิมเมจสแกนเนอร์แบบต่างๆ ดังที่แสดงไว้ในข้างต้น เราจะเห็นได้ว่าแบบที่เหมาะสมกับความต้องการ ก็คือ แบบ เทเบิลอิมเมจสแกนเนอร์(ใช้เซ็นเซอร์ชนิดใช้กับเลนส์) เพราะเหมาะกับการใช้อ่านภาพแผ่นโลหะที่มีความหนาและมีขนาดใหญ่ ดังนั้นแนวทางในการพัฒนาเราจะนำลักษณะ ตลอดจนวิธีการต่างๆที่มีในแบบนี้มาใช้ตามความเหมาะสม

2.3 <u>การติดต่อกับผู้ใช้</u>

ส่วนนี้มีไว้ เพื่อทำให้ผู้ใช้สามารถควบคุมการทำงานของเครื่องอิมเมจสแกนเนอร์ได้สะดวกจากเครื่องคอมพิวเตอร์ อีกทั้งยังเอื้อประโยชน์ในการดึงข้อมูลที่ได้ จากการอ่านภาพของเครื่องอิมเมจสแกนเนอร์ มาใช้กับโปรแกรมบนเครื่องไมโครคอมพิวเตอร์ได้ง่าย

2.3.1 การเชื่อมต่อทางฮาร์ดแวร์ระหว่างเครื่องอิมเมจสแกนเนอร์กับเครื่องไมโครคอมพิวเตอร์

การควบคุมระบบการทำงานของอิมเมจสแกนเนอร์ จำเป็นต้องใช้งานร่วมกับผู้ใช้ โดยผ่านไมโครคอมพิวเตอร์นั้น จำเป็นที่จะต้องมีระบบการเชื่อมต่อระหว่างกัน ซึ่งก็มีด้วยกันหลายทาง เช่น RS-232, พอร์ตขนาน และ SCSI

RS-232 เป็นการสื่อสารแบบอนุกรม สามารถส่งสัญญาณไปได้ไกล ใช้สายสัญญาณน้อย และ มีอยู่ในไมโครคอมพิวเตอร์โดยทั่วไป แต่เนื่องจากความเร็วในการส่งข้อมูลต่ำจึงไม่เหมาะที่จะนำมาใช้กับงานวิจัยนี้

นอร์ดขนาน เป็นการสื่อสารแบบขนาน สามารถส่งสัญญาณได้ไม่ไกลนัก ใช้สายสัญญาณมากกว่าแบบอนุกรม และมีอยู่ในไมโครคอมพิวเตอร์โดยทั่วไป ความเร็วในการส่งข้อมูลสูงกว่าแบบอนุกรม

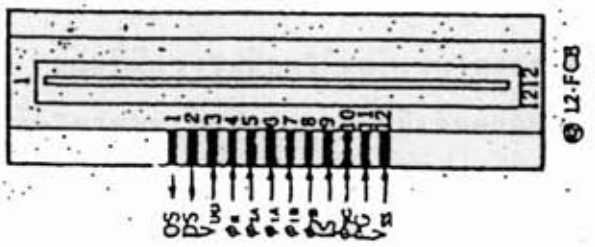
SCSI เป็นการสื่อสารแบบขนานที่มีความเร็วสูงมาก มักนำมาใช้กับฮาร์ดดิสก์ที่มีความจุข้อมูลสูง โดยใช้สายสัญญาณมากกว่าทุกๆแบบ แต่ไม่มีใช้ในเครื่องไมโครคอมพิวเตอร์โดยทั่วไป และ ส่งสัญญาณไปได้ไม่ไกลนัก

จากลักษณะและคุณสมบัติต่างๆของแต่ละแบบ จะเห็นว่าแบบ RS-232 นั้นความเร็วไม่เหมาะสมกับการส่งข้อมูลภาพซึ่งมีจำนวนมาก เพราะจะใช้เวลามากในการส่งข้อมูลแบบพอร์ดขนานนั้น จะต้องมีวิธีการลดจำนวนข้อมูล เพื่อให้ความเร็วเป็นไปตามต้องการ ส่วนแบบท้ายสุดแม้ว่าจะมีความเร็วสูง แต่ก็ไม่มีใช้กันโดยทั่วไป และยังใช้กับสายสัญญาณสั้นๆเท่านั้น ดังนั้นจึงได้เลือกแบบพอร์ดขนานมาใช้งานเพราะเหมาะสมกว่าแบบอื่นๆ

2.4 ลักษณะและการทำงานของอิมเมจเซ็นเซอร์(Image Sensor) [6][7][8][9]

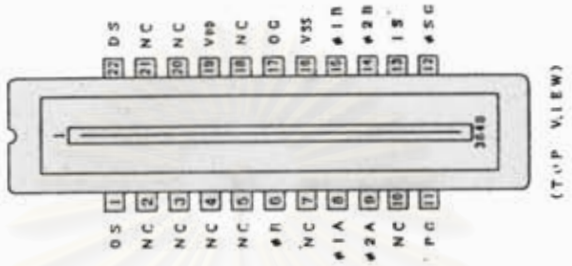
อิมเมจเซ็นเซอร์ที่ใช้ในการอ่านภาพ ที่นิยมใช้กันในปัจจุบันจะเป็นชนิด CCD สามารถแบ่งตามลักษณะการใช้งานได้เป็น 2 ชนิด

- ชนิดคอนแท็ค มีขนาดใหญ่กว่าชนิดใช้กับเลนส์ การใช้งานต้องนำเซ็นเซอร์ไปวางชิดติดกับภาพเป้าหมาย อิมเมจเซ็นเซอร์ชนิดนี้นิยมใช้ทำเครื่องโทรสาร ดังนั้นจึงต้องมีความยาวเท่ากับความกว้างของกระดาษ หรือรูปที่ต้องการอ่าน ข้อเสียคือไม่สามารถทำยาวมากๆได้ การนำมาต่อกันก็ทำได้ยาก ดังแสดงในรูปที่ 2.4



รูปที่ 2.4 ตัวเซ็นเซอร์ชนิดคอนแท็ค

- ชนิดใช้กับเลนส์ มีขนาดเล็ก การใช้งานจะต้องใช้เลนส์ในการลดขนาดของภาพเป้าหมายลงบนตัวเซ็นเซอร์ เซ็นเซอร์ชนิดนี้จะมีความละเอียดสูงมาก การนำมาต่อกันทำได้โดยการต่อเหลื่อมกัน ดังแสดงในรูปที่ 2.5



รูปที่ 2.5 ตัวเซ็นเซอร์ชนิดใช้กับเลนส์

2.4.1 การทำงานของตัวอิมเมจเซ็นเซอร์ แบ่งออกเป็นสามขั้นตอน

1) การกำเนิดอิเล็กตรอนจากแสงตกกระทบ

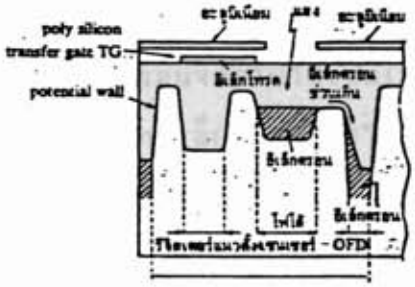
แสงที่เข้ามาจะถูกอีลีเมนต์ภายในตัวเซ็นเซอร์ เปลี่ยนให้เป็นอิเล็กตรอน และปริมาณของมันจะเป็นปฏิภาคต่อความเข้มของแสง แผ่นซิลค์อะลูมินั่ม(AI)ของแต่ละอีลีเมนต์(ดังแสดงในรูปที่ 2.6 ถึง รูปที่ 2.9) ทำหน้าที่กำบังแสงไม่ให้แผ่กระจายไปบริเวณอื่น นอกจากหน่วยโฟโต้เซ็นเซอร์

2) การเก็บอิเล็กตรอน (Store)

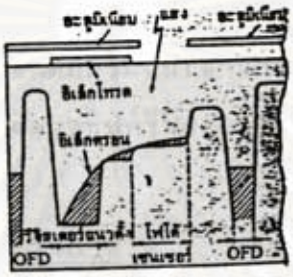
บ่อแรงดันแต่ละหน่วย(potential well) ดังแสดงในรูปที่ 2.6 ที่เกิดขึ้นภายในนั้น จะเห็นว่าอิเล็กตรอนเหล่านี้จะถูกเก็บอยู่ในบ่อ และ ถึงแม้บางครั้งมีปริมาณเกิน อันเนื่องมาจากแสงสว่างจ้ามากมาตกกระทบก็ตาม แต่ตัวชิพเซ็นเซอร์สามารถจัดการได้ โดยอิเล็กตรอนที่ล้นจะออกไปทางระบาย O.F.D.(overdrain)

3) การถ่ายเทอิเล็กตรอน (Electronic Transfer)

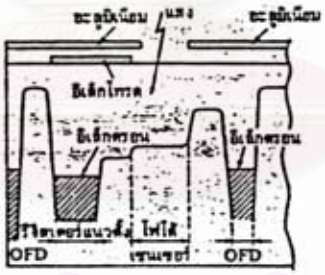
ขณะที่มีแรงไฟป้อนให้แก่อิเล็กโทรดของอีลีเมนต์ ความลึกของบ่อแรงดันใต้อิเล็กโทรดจะเพิ่มขึ้น ฉะนั้นอิเล็กตรอนที่ถูกกักเก็บไว้นั้นจะเริ่มถ่ายลงมาให้กับเวอร์ติคอล รีจิสเตอร์(รูปที่ 2.6 และที่ 2.7) ลักษณะเช่นนี้เหมือนกับประตูเขื่อนถูกเปิดออกขณะที่จำนวนของอิเล็กตรอนถูกถ่ายออกหมดแล้ว ดังรูปที่ 2.9 ไฟที่ป้อนให้กับอิเล็กโทรดจะหยุดด้วย ดังแสดงในรูปที่ 2.9



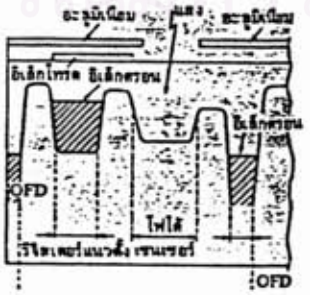
รูปที่ 2.6 แสดงการถ่ายเทอิเล็กตรอนออกจากแอ่งเก็บเมื่อแสงมีมากเกินไป



รูปที่ 2.7 แสดงการถ่ายเทอิเล็กตรอนไปสู่รีจิสเตอร์



รูปที่ 2.8 ถ่ายเทเสร็จแล้ว



รูปที่ 2.9 เตรียมพร้อมส่งต่อไป

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

นั่นคือ ความลึกของบ่อ ก็จะลดน้อยหรือตื้นกลับคืนมาเหมือนเดิม ด้วยเหตุนี้เอง การถ่ายเทอิเล็กตรอนจากโฟโต้เซนเซอร์ไปสู่เวอร์ติคอลรีจิสเตอร์ได้สิ้นสุดลง ระหว่างการถ่ายเทอิเล็กตรอนภายในอยู่ จะยังไม่มีการกำเนิดอิเล็กตรอนใหม่ ถึงแม้ว่าโฟโต้เซนเซอร์ขณะนี้กำลังรับแสงอยู่ก็ตาม นี่ก็เพราะว่าความเร็วในการถ่ายเทนั้นสูงกว่า การเกิดอิเล็กตรอนสรุปได้ว่าคุณสมบัติของตัวอิมเมจเซ็นเซอร์ ปราศจากปัญหาเกี่ยวกับการเผาไหม้ และ ภาพเป็นดาวหางเป็นต้น

ทั้งสามขั้นตอนนี้ เป็นการแสดงพื้นฐานการทำงานของตัวอิมเมจเซ็นเซอร์ ทีแรกมันรับแสงสว่างที่ตกกระทบเข้ามาแล้วกำเนิดเป็นกระแสอิเล็กตรอนที่เก็บไว้นั้นถ่ายเทผ่านเวอร์ติคอลชิฟท์รีจิสเตอร์และฮอริซอนทอลรีจิสเตอร์ แล้วส่งไปยังวงจรเอาต์พุต

2.5 <u>องค์ประกอบของการอ่านภาพ</u>

นอกจากองค์ประกอบของเครื่องอ่านภาพแล้ว อีกส่วนก็คือสภานของการอ่านภาพต่างๆ เช่น ระดับความเข้มแสง มุมและตำแหน่งที่รับภาพ ก็ยังอุปกรณ์ที่เกี่ยวข้อง เช่น เลนส์ ซึ่งมีอยู่หลายแบบด้วยกัน ดังนั้นจึงต้องเลือกให้เหมาะสมกับสภาพงาน

2.5.1 เลนส์ [10][11]

จะถูกนำมาใช้กรณีที่ตัวเซ็นเซอร์เป็นชนิดที่ใช้กับเลนส์ ในปัจจุบันนี้ได้มีการพัฒนาและ แก้ไขข้อเสียต่าง ๆ ของเลนส์ เช่น การผิดเพี้ยนที่เกิดบริเวณขอบของภาพ และ ความคมชัดของภาพ เป็นต้น โดยการใช้วัสดุที่ดีมาทำเลนส์ และการนำเลนส์มาซ้อนกันเป็นชุดแทนการใช้เลนส์เดี่ยว เลนส์เหล่านี้สามารถหาซื้อได้ง่ายในปัจจุบัน และ ถูกใช้ในงานการถ่ายภาพทั่วไป ซึ่งจะได้อธิบายเลนส์แบบต่าง ๆ ดังต่อไปนี้

เลนส์มุมกว้าง ลักษณะพิเศษของแบบนี้ คือ สามารถครอบคุมเนื้อที่ได้มาก โดยใช้ระยะโฟกัสไม่มาก แต่ภาพที่ได้จากเลนส์แบบนี้มีความผิดเพี้ยนค่อนข้างมาก

เลนส์นอร์มอล เลนส์แบบนี้ ภาพที่ได้จะมีลักษณะใกล้เคียงการมองของตาคนเรา ภาพที่ได้จากเลนส์แบบนี้มีความผิดเพี้ยนค่อนข้างน้อย

เลนส์ไมโคร ลักษณะพิเศษของแบบนี้ คือ เหมาะกับการจับภาพในระยะใกล้ในขณะที่มีความผิดเพี้ยนน้อยกว่าแบบอื่นๆ แต่ครอบคลุมเนื้อที่น้อยกว่า

แบบอื่นๆ

2.5.2 ทิศทางของแสงและผลที่ได้ [5]

อิมเมจสแกนเนอร์จะต้องมีแหล่งกำเนิดแสง โดยทิศทางของแสงนั้น สามารถเปลี่ยนแปลงลักษณะของภาพได้ ซึ่งจะได้อธิบายถึงลักษณะแบบต่างๆ ดังต่อไปนี้

ส่องมาทางด้านหน้าวัตถุ ทำให้สามารถมองเห็นระดับโทนแสง และรายละเอียดบนผิวของวัตถุด้วย

ส่องมาทางด้านข้างวัตถุ ช่วยทำให้เห็นความหนาบางของวัตถุ ภาพที่ได้ดูมีความลึก ทั้งยังมีระดับโทนแสง และรายละเอียดบนผิวของวัตถุอีกด้วย

ส่องมาทางด้านหลังวัตถุ ข้อมูลที่ได้จากวิธีนี้จะไม่ซับซ้อน สามารถกำจัด Gray Scale ดังนั้นภาพที่ได้จะทึบ คือ มองเห็นเพียงขอบของวัตถุแต่จะไม่เห็นรายละเอียดบนผิวของวัตถุ มีประโยชน์ในการลดจำนวนข้อมูลภาพ

ทิศทางของแสงที่เหมาะสม กับ การนำมาใช้ แบบที่แสงส่องมาทางด้านข้างนั้นไม่เหมาะสมเพราะเราต้องการความสม่ำเสมอของแสง และ ไม่ต้องการความลึกของภาพ ส่วนแบบแสงส่องมาทางหลังนั้น แม้จะให้ข้อดีกับการลดจำนวนข้อมูลก็ตาม แต่เนื่องจากเราจะนำไปใช้กับแผ่นโลหะซึ่งมีความหนา ทำให้ขนาดบริเวณปลายๆ ของขอบภาพผิดเพี้ยนไป ดังนั้นแบบที่มีความเหมาะสมที่สุดก็คือแบบที่แสงส่องมาทางด้านหน้าวัตถุ แบบนี้มีระดับโทนแสงซึ่งไม่ต้องการเพราะข้อมูลที่ได้จะมากเกินไป ส่งผลทำให้การทำงานของระบบช้าลง ดังนั้นเราจะต้องลดจำนวนข้อมูล โดยการแบ่งความเข้มของแสงออกเป็นสองระดับเท่านั้น ซึ่งมีแนวคิดวิธีการในการลดจำนวนข้อมูล ดังจะได้แสดงในบทที่ 3 ต่อไป

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 3

แนวความคิดในการออกแบบ

ก่อนที่จะทำการออกแบบสร้างระบบอิมเมจสแกนเนอร์นั้น เราจะต้องให้ข้อกำหนด ทั้งนี้เพื่อเป็นขอบเขตของการวิจัย และหลังจากนั้นจะได้กล่าวถึงการออกแบบต่อไป

3.1 ข้อกำหนดของเครื่องอ่านภาพแผ่นโลหะเรียบ

1) เนื้อที่ของการสแกน

กว้าง 1 เมตร ยาว 2.4 เมตร

2) ความแม่นยำ

0.2 มิลลิเมตร

3) การต่อใช้งาน

ติดต่อผ่านทางพอร์ตขนาน

4) เวลาที่ใช้ในการอ่าน

ไม่เกิน 20 นาที

จากการศึกษาในบทที่ 2 เป็นพื้นฐานสำหรับนำมาใช้เป็นแนวความคิดในบทนี้ ซึ่งเป็นการโครงร่างคร่าวๆของระบบที่จะสร้างขึ้น ดังต่อไปนี้

3.2 ส่วนประกอบฮาร์ดแวร์ของเครื่องอ่านภาพแผ่นโลหะเรียบ

ส่วนนี้เป็นส่วนที่เกี่ยวกับกระบวนการเพื่อให้ได้ข้อมูลจากภาพแผ่นโลหะ ซึ่งแบ่งออกได้เป็นส่วน ๆ ดังนี้

1) ส่วนโต๊ะวาง เป็นส่วนที่ใช้สำหรับวางแผ่นโลหะ ซึ่งลักษณะของโต๊ะขึ้นกับทิศทางของแสง กรณีที่เราใช้แสงผ่านแผ่นโลหะมายังตัวเซ็นเซอร์โดยตรงนั้นตัวพื้นโต๊ะที่ใช้จะต้องยอมให้แสงผ่านได้ แต่ถ้าเป็นกรณีที่ใช้แสงสะท้อนแผ่นโลหะตัวพื้นโต๊ะที่ใช้จะต้องไม่ท้อนแสง

2) ส่วนหัวอ่าน เป็นส่วนที่ควบคุมกระบวนการอ่านข้อมูล ตั้งแต่ตัวเซ็นเซอร์จนถึงการส่งข้อมูล รวมทั้งควบคุมการเคลื่อนไหวของหัวอ่านด้วยภายใต้การควบคุมของไมโครโปรเซสเซอร์ ซึ่งส่วนประกอบในระบบหลักๆมีดัง

ต่อไปนี้

- เซ็นเซอร์
- เลนส์ (กรณีที่ตัวเซ็นเซอร์ชนิดที่ใช้กับเลนส์)
- แหล่งกำเนิดแสง
- วงจรขยายสัญญาณวิดีโอ
- วงจรควบคุมการทำงานของตัวเซ็นเซอร์
- วงจรแปลงสัญญาณวิดีโอเป็นดิจิตอล
- วงจรการรับส่งข้อมูล
- วงจรไมโครโปรเซสเซอร์

3) ส่วนการเคลื่อนไหว เนื่องจากภาพที่ต้องการอ่านนั้นมีความยาว และ ตัวเซ็นเซอร์ก็มีลักษณะเป็นเส้นเพียงเส้นเดียว ดังนั้นการอ่านภาพให้ได้ทั้งหมดจะต้องมีการเลื่อนตำแหน่งของการอ่าน ซึ่งส่วนหลัก ๆ มีดังนี้

- กลไกการเคลื่อนไหว ชุด linear motion[12][13]
- ชุดมอเตอร์ขับนำ [14]

บล็อกไดอะแกรมของระบบแบ่งออกเป็น 6 ส่วน(ดังแสดงในรูปที่3.1)

1) ส่วนตัวอ่านภาพ (Linear Image Sensor)

อิมเมจเซ็นเซอร์ที่จะนำมาใช้เป็นแบบ CCD ซึ่งทำหน้าที่อ่านภาพ ซึ่งเป็นสัญญาณอานาลอกของแต่ละจุดภาพเรียงกันเข้ามาในระบบ

2) ส่วนการอ่านข้อมูลภาพ (Image Data Circuit)

ทำหน้าที่ควบคุมการทำงานของหัวอ่านข้อมูล ให้สามารถอ่านข้อมูลได้อย่างถูกต้อง และทำการแปลงข้อมูลให้เป็นรูปแบบที่สามารถนำไปใช้ในการระบุสถานะของข้อมูล และ บ่งถึงตำแหน่งของข้อมูลด้วย

3) ส่วนการเลื่อนตำแหน่งหัวอ่าน (Head-reader Movement)

เป็นส่วนที่ทำหน้าที่ควบคุมการเคลื่อนไหว โดยการส่งสัญญาณไฟฟ้าไปควบคุมสเต็บปิ้งมอเตอร์ ให้หมุนเลื่อนหัวอ่านไปยังตำแหน่งที่ต้องการอ่านภาพ

4) ส่วนตรวจสอบตำแหน่งเริ่มต้น (Zero Reference)

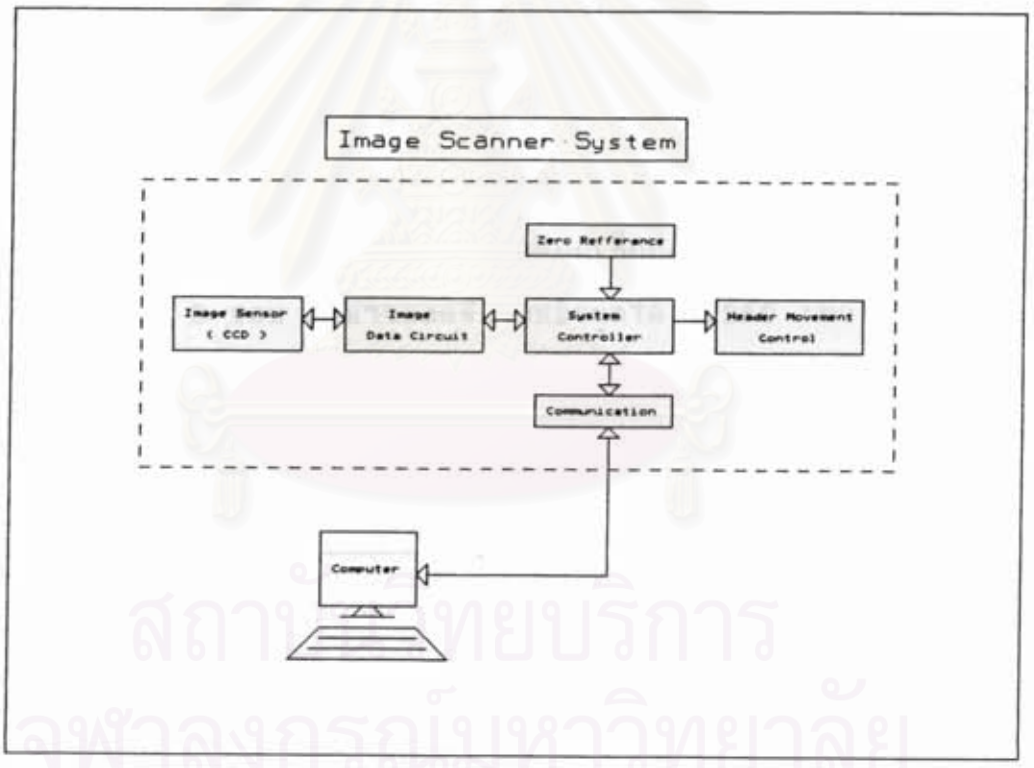
ทำหน้าที่ตรวจสอบหาตำแหน่งเริ่มต้นของการอ่านภาพ โดยใช้ไมโครสวิตช์ ในการตรวจหาตำแหน่งอ้างอิงนั้น

5) ส่วนการติดต่อสื่อสาร (communication)

ทำหน้าที่ในการเชื่อมต่อเครื่องอ่านภาพแผ่นโลหะเรียบ กับเครื่องไมโครคอมพิวเตอร์ เพื่อควบคุมการทำงานของระบบการสแกน

6) ส่วนการควบคุมระบบ (System Controller)

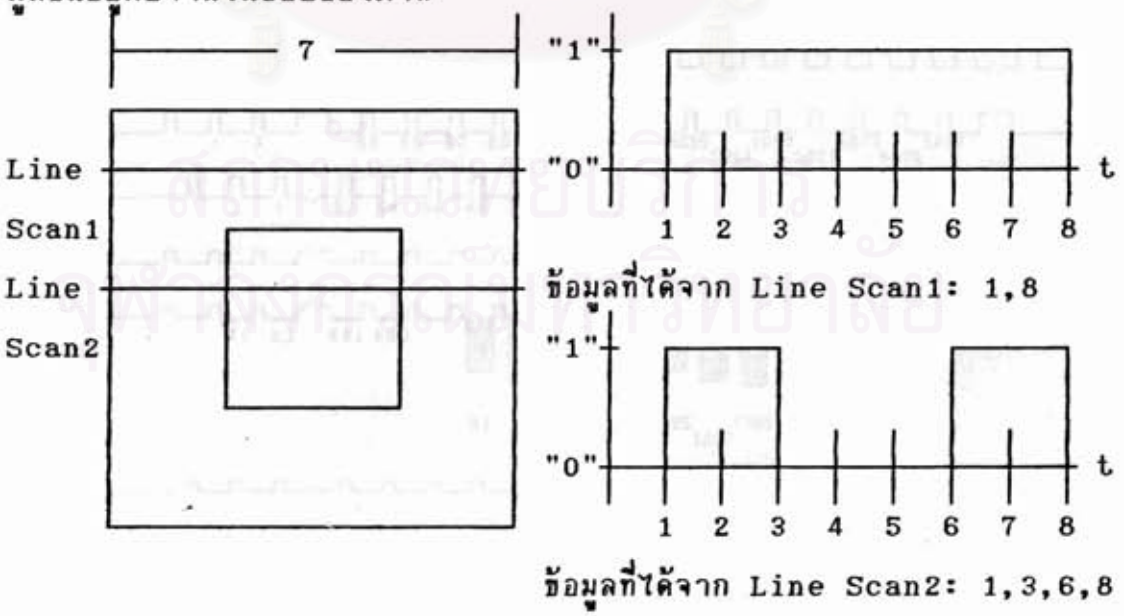
เป็นส่วนที่ทำหน้าควบคุมการทำงานของเครื่องอ่านภาพแผ่นโลหะเรียบทั้งระบบ ให้การทำงานของแต่ละส่วนสอดคล้องกันและสามารถควบคุมกระบวนการทั้งหมดในการปฏิบัติการโดยเครื่องไมโครคอมพิวเตอร์



รูปที่ 3.1 ระบบการอ่านภาพแผ่นโลหะเรียบ

3.3 เทคนิคการลดจำนวนข้อมูลจากการสแกนภาพ

เนื่องจากความต้องการให้ระบบมีความเร็วการสแกนภาพสูง และใช้เนื้อที่เก็บข้อมูลน้อย ซึ่งจากลักษณะของข้อมูล และความต้องการข้อมูลส่วนที่เป็นขอบของวัตถุเท่านั้น ดังนั้นจึงนำวิธีการลดข้อมูลมาใช้ ดังแสดงในรูปที่ 3.2 ส่วนฉากหลังจะเป็นสีดำ ภาพที่ได้ในส่วนนี้มีระดับความเข้มแสงสะท้อนกลับต่ำข้อมูลที่ได้ส่วนนี้จะแทนเท่ากับ '0' ส่วนข้อมูลที่ได้จากส่วนที่เป็นวัตถุนั้นระดับความเข้มแสงสะท้อนกลับสูงข้อมูลที่ได้ในส่วนนี้จะแทนเท่ากับ '1' ดังนั้นก็จะได้รูปสัญญาณเป็นสัญญาณดิจิตอล ซึ่งเมื่อพิจารณาจะเห็นว่าการเก็บข้อมูลนี้ไม่จำเป็นต้องเก็บทั้งหมด แต่จะเลือกเก็บข้อมูลเฉพาะส่วนที่มีการเปลี่ยนแปลงเท่านั้น และเราเรียกส่วนนี้ว่า "ข้อมูลการเปลี่ยนแปลง"(transition Data) การลดจำนวนข้อมูลโดยวิธีนี้สามารถลดจำนวนข้อมูลได้หลายเท่าตัว กรณีมีจุดรับภาพ 5,000 จุด ถ้าเป็นแบบเก็บข้อมูลทุกๆจุดจะใช้เนื้อที่การเก็บ 5,000 ไบต์ แต่ถ้าเป็นแบบที่เก็บข้อมูลเฉพาะที่เป็นขอบจะใช้เนื้อที่การเก็บขอบละ 2 ไบต์ สมมุติถ้ามี 2 ขอบ จะใช้เนื้อที่การเก็บ 4 ไบต์ และถ้ามี 4 ขอบ จะใช้เนื้อที่การเก็บ 8 ไบต์ เมื่อเปรียบเทียบระหว่างแบบเก็บทุกๆจุดกับแบบที่เก็บข้อมูลเฉพาะที่เป็นขอบแล้ว 2 ขอบ สามารถลดจำนวนข้อมูลได้ 1,250 เท่า และ ที่ 4 ขอบ สามารถลดจำนวนข้อมูลได้ 650 เท่า จะเห็นได้ว่าอัตราส่วนการลดจำนวนข้อมูลขึ้นอยู่กับจำนวนขอบของภาพ

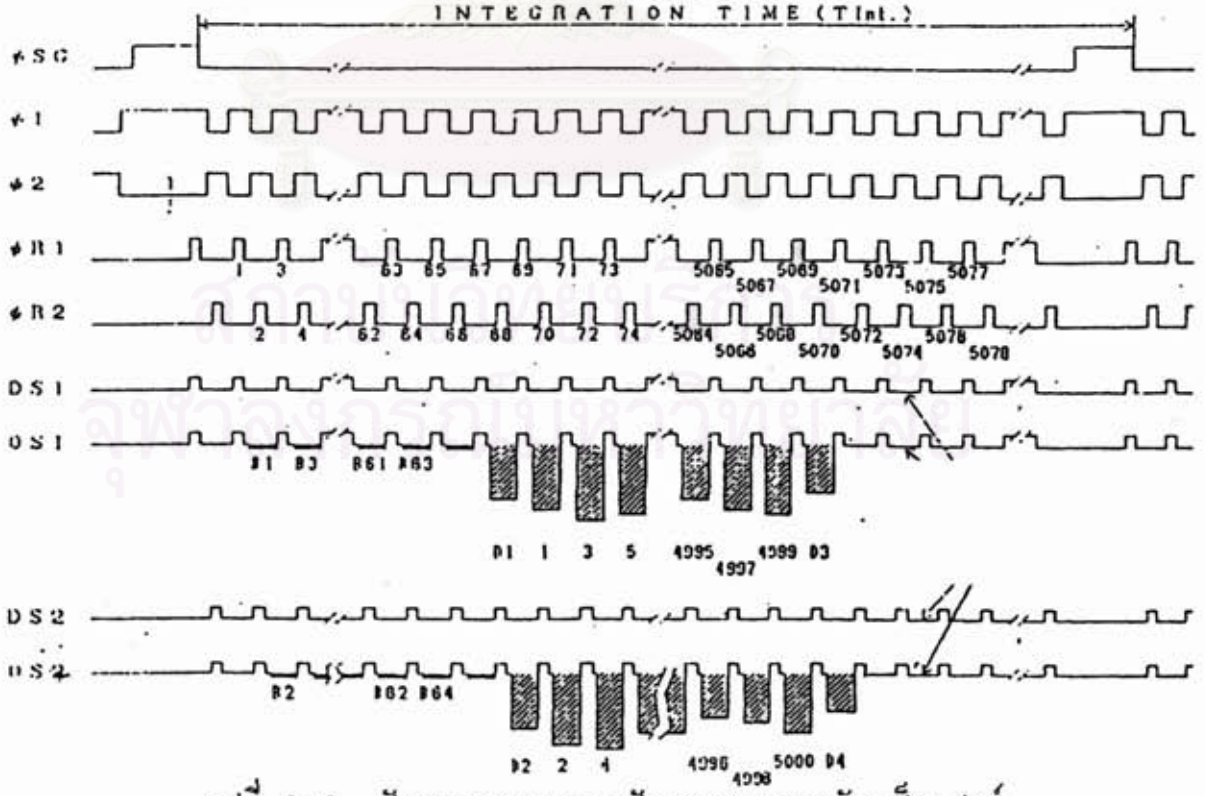


รูปที่ 3.2 ข้อมูลที่ได้จากการสแกนอ่านแต่ละเส้น

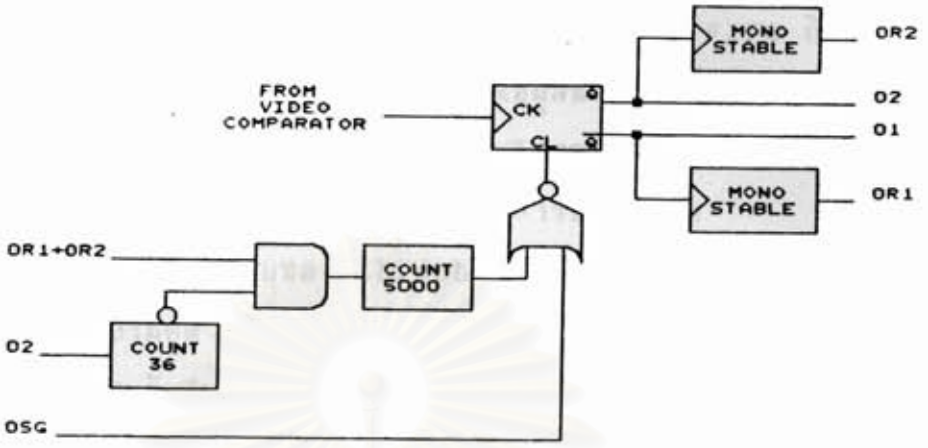
3.4 การควบคุมการอ่านสัญญาณภาพจากอิมเมจเซ็นเซอร์ [15]

การออกแบบในส่วนนี้จำเป็นจะต้องรู้เสียก่อนว่า ตัวอิมเมจเซ็นเซอร์นั้นมีลำดับขั้นตอนของการควบคุมอย่างไร เพื่อให้ได้มาซึ่งสัญญาณภาพมาจากตัวเซ็นเซอร์ ดังนั้นเราจะต้องมาศึกษาไทม์มิ่งของตัวเซ็นเซอร์ก่อน ดังแสดงในรูปที่ 3.3

เมื่อเรานำอิมเมจเซ็นเซอร์ไปรับภาพ ซึ่งจุดรับภาพที่อยู่ภายในจะประจุสัญญาณ จากนั้นก็จะให้สัญญาณ Φ_{SG} เพื่อเคลื่อนย้ายประจุจากจุดรับภาพไปยังรีจิสเตอร์ของแต่ละจุดรับภาพซึ่งแยกเป็น 2 ชุด คือ จุดคี่และจุดคู่ โดยใช้ Φ_1 และ Φ_2 ที่มีความต่างเฟสกัน 180 องศา และมี Φ_{R1},Φ_{R2} เป็นตัวเปิดเกทให้สัญญาณ Φ_{S1},Φ_{S2} ซึ่งมีสัญญาณ Φ_{R1} และ Φ_{R2} ผสมอยู่ ดังนั้นเราจะต้องนำเอาสัญญาณ DS_1,DS_2 มาหักล้างออกไป เพื่อให้สัญญาณที่ออกไปสู่วงจรขยายจะมีเพียงสัญญาณภาพเท่านั้น การควบคุมการทำงานของตัวอิมเมจเซ็นเซอร์นั้นจะต้องเป็นวงรอบ โดยวงรอบจะเริ่มนับหลังจากที่ให้สัญญาณ Φ_{SG} และ Φ_{R2} ครบ 36 ไซเคิล หลังจากนั้นก็จะใช้ไทม์เมอร์เคาน์เตอร์นับจนครบ 5,000 จุด เพื่อเลื่อนข้อมูลที่อยู่ภายในตัวอิมเมจเซ็นเซอร์ออกไปและเริ่มวงรอบใหม่ บล็อคไดอะแกรมการสร้างสัญญาณควบคุมดังแสดงในรูปที่ 3.4



รูปที่ 3.3 สัญญาณควบคุมและสัญญาณภาพจากตัวเซ็นเซอร์



รูปที่ 3.4 บล็อคไดอะแกรมการสร้างสัญญาณควบคุม

3.5 บล็อคไดอะแกรมการอ่านภาพ

เป็นการอธิบายถึงการเปลี่ยนแปลงของข้อมูล โดยมองระบบการทำงานแยกออกเป็นโมดูล ซึ่งจะสนใจแต่เฉพาะอินพุตและเอาต์พุตเท่านั้น ดังที่แสดงใน รูปที่ 3.5

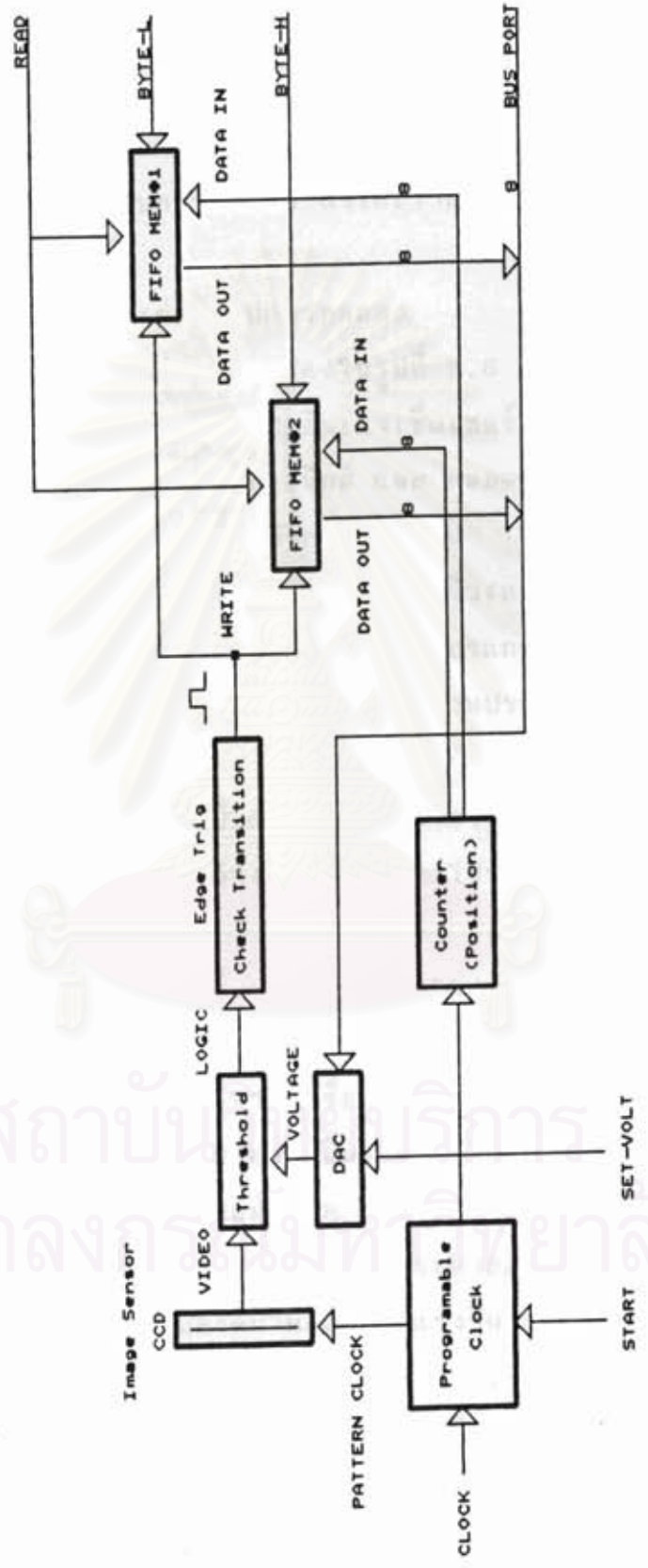
จากรูปจะเห็นได้ว่าส่วนที่เป็นอินพุตที่มาจากภายนอก มี 4 สัญญาณ คือ สัญญาณวิดีโอ,clock และ สัญญาณ start ซึ่งสัญญาณวิดีโอนั้นมาจากอิมเมจเซ็นเซอร์ที่ใช้เป็นตัวรับสัญญาณภาพ ส่วน programable clock เป็นโมดูลที่ใช้สำหรับสร้างสัญญาณควบคุมการอ่านภาพจากตัวอิมเมจเซ็นเซอร์ และส่งสัญญาณให้กับโมดูล counter เพื่อบอกตำแหน่งของข้อมูลที่อ่านได้

เริ่มต้นการทำงานโดยการให้สัญญาณ start เพื่อทำให้โมดูล programable clock เริ่มทำการสร้างสัญญาณควบคุมการอ่านข้อมูลจากตัวอิมเมจเซ็นเซอร์ โดยทำงานตามจังหวะของสัญญาณ clock จากการให้สัญญาณ pattern กับตัวอิมเมจเซ็นเซอร์ ทำให้มีสัญญาณวิดีโอออกมาจากตัวอิมเมจเซ็นเซอร์ และป้อนสัญญาณวิดีโอที่ได้เข้าไปยังโมดูล Threshold จะได้เอาต์พุตออกมาเป็นสัญญาณโลจิก และส่งต่อให้กับโมดูล check transition เพื่อทำการตรวจการเปลี่ยนแปลงของระดับลอจิก ทำให้ได้เอาต์พุตออกมาเป็นสัญญาณการเปลี่ยนแปลง(transition signal) โดยจะนำเอาสัญญาณการเปลี่ยนแปลงมาทำการเลือกตำแหน่งของข้อมูลจากโมดูล counter เข้ามาเก็บไว้ในหน่วยความจำ FIFO (First In Fist Out) แต่เนื่องจากข้อมูลตำแหน่งนั้นมี

มากกว่า 256 จุด ดังนั้นจึงต้องใช้ FIFO ที่รับอินพุตได้มากถึง 16 บิท แต่โดยปกติ FIFO จะรับอินพุตได้เพียง 8 บิท ทำให้ต้องใช้ FIFO ถึง 2 ตัวโดยหน่วยความจำ FIFO นั้น มีไว้สำหรับเก็บข้อมูลชั่วคราวที่ความเร็วสูง หลังจากนั้นก็รอการอ่านข้อมูลจากบอร์ดไมโครคอนโทรลเลอร์ ซึ่งจะใช้สัญญาณ READ,BYTE-L และ BYTE-H ในการควบคุมการอ่านข้อมูล

ส่วนการปรับระดับของ threshold นั้น ทำได้ด้วยการเปลี่ยนข้อมูลระดับที่ให้กับโมดูล DAC(Digital to Analog Converter) โดยการส่งค่าสัญญาณระดับไปที่พอร์ต BUS PORT หลังจากนั้นจึงทำการส่งสัญญาณควบคุม SET-VOLT เพื่อเปลี่ยนค่าเป็นระดับที่ต้องการ

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 3.5 บล็อคไดอะแกรมการอ่านภาพ

3.6 การทดลองเพื่อหาวิธีการที่เหมาะสม

จากการออกแบบทดลองสร้างระบบการอ่านภาพแผ่นโลหะ โดยใช้กล้องเพียงตัวเดียวจับภาพแผ่นโลหะยาวประมาณ 1 เมตร ซึ่งมีการเจาะรูสี่เหลี่ยมกว้างขนาด 25 มิลลิเมตร เจาะเว้นระยะ 10 มิลลิเมตร มีจำนวน 28 รู นำมาใช้เป็นแบบสำหรับการทดสอบ ทำให้เราสามารถรู้ถึงปัญหาที่เกิดขึ้น และหาวิธีแก้ไขปัญหาที่เกิดขึ้น

3.6.1 วงจรต่างๆในการทดลอง

ส่วนต่างๆ ดังแสดงในรูปที่ 3.6 ซึ่งประกอบด้วย บอร์ดซีพียู, บอร์ดอินเตอร์เฟส, วิดีโอแอมป์, อิมเมจเซ็นเซอร์ กับ เลนส์ ซึ่งรวมกันเป็น กล้องจับภาพ, บัลลาสต์อิเล็กทรอนิกส์ และ หลอดฟลูออเรสเซ็นต์ เพื่อให้แสงสว่างกับแผ่นโลหะที่ต้องการจะทำการอ่านภาพ โดยข้อมูลที่ได้จากกล้องจะส่งผ่านบอร์ดอินเตอร์เฟสไปยังเครื่องไมโครคอมพิวเตอร์ทางพอร์ตขนานและส่วนรอมอีมูเลเตอร์นั้น มีไว้สำหรับการดาวน์โหลดโปรแกรมให้กับบอร์ดซีพียู ทำให้ง่ายต่อการทดลอง และการแก้ไขปรับปรุง ซึ่งส่วนประกอบต่างๆของการทดลองมี 5 ส่วน ดังต่อไปนี้

1) บอร์ดซีพียู เป็นส่วนที่ใช้ในการควบคุมระบบของเครื่องอ่านภาพแผ่นโลหะ ซึ่งการกำหนดการทำงานได้ด้วยการโปรแกรมรอม หรือ อีพรอมแล้วนำมาใส่ในบอร์ดนี้

2) รอมอีมูเลเตอร์ ทำหน้าที่จำลองเป็น รอม เพื่อช่วยให้การพัฒนาโปรแกรมเป็นไปโดยสะดวก ทั้งนี้เพราะมันสามารถที่จะเปลี่ยนแปลงโปรแกรมได้ โดยการดาวน์โหลดจากเครื่องไมโครคอมพิวเตอร์ ซึ่งทำให้ไม่ต้องเสียเวลาในการโปรแกรม รอม หรือ อีพรอม และเมื่อโปรแกรมเป็นที่แน่นอนแล้ว จึงจะโปรแกรมลงบน รอม หรือ อีพรอม ในการนำไปใช้งานจริง

3) วิดีโอแอมป์ ทำหน้าที่ในการขยายสัญญาณที่ได้รับมาจากอิมเมจเซ็นเซอร์ ซึ่งเสียบอยู่บนบอร์ดนี้ให้มีสัญญาณแรงขึ้น และส่งสัญญาณมาที่บอร์ดอินเตอร์เฟส

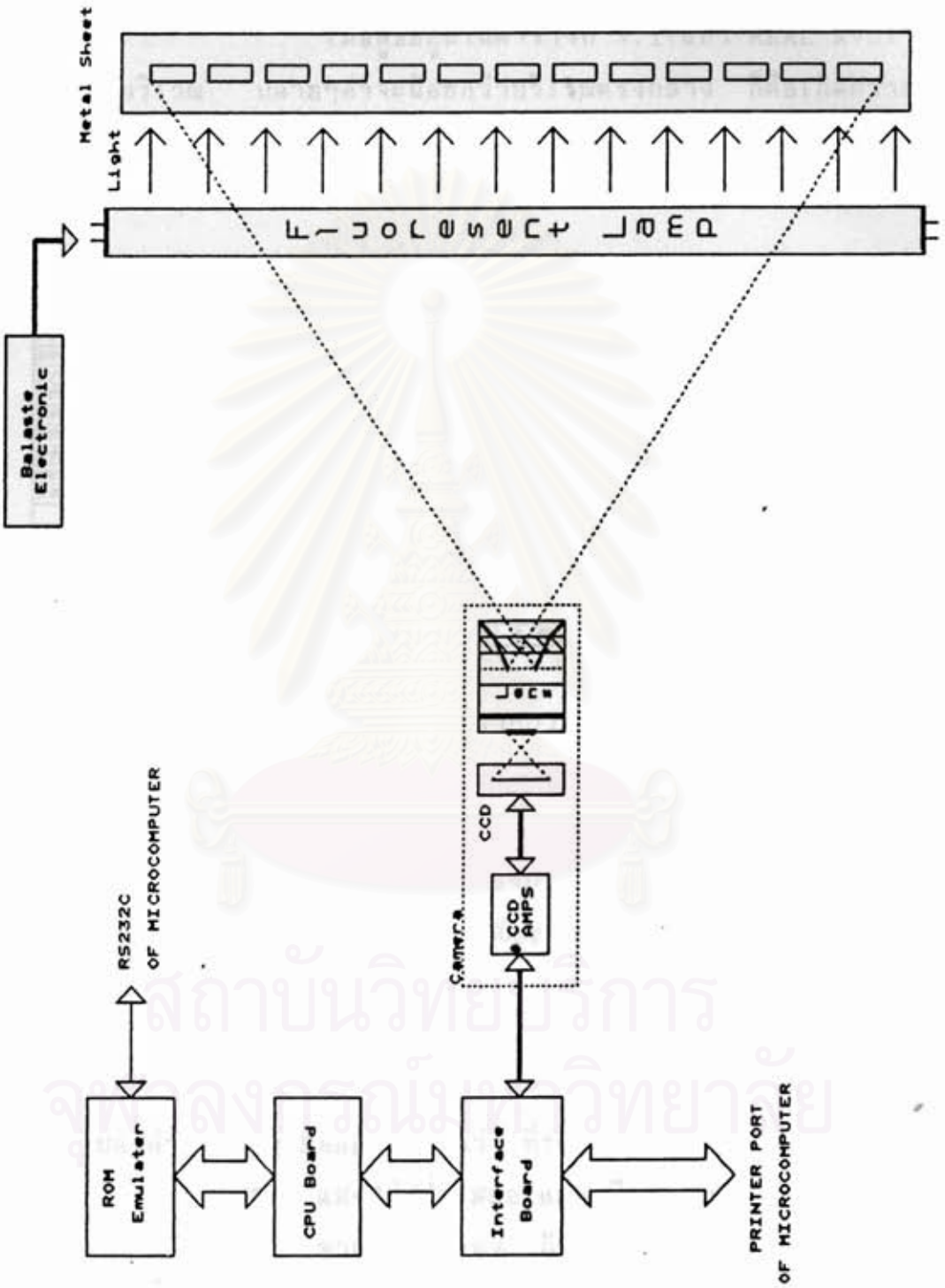
4) บอร์ดอินเตอร์เฟส เป็นส่วนที่เชื่อมต่อการควบคุมกับบอร์ดซีพียู, การติดต่อสื่อสาร กับ เครื่องไมโครคอมพิวเตอร์ และยังรับสัญญาณควบคุมอิมเมจเซ็นเซอร์ รวมทั้งนำสัญญาณที่ได้จากอิมเมจเซ็นเซอร์ ซึ่งผ่านการขยายสัญญาณแล้ว เปลี่ยนรูปแบบจากสัญญาณอานาลอกไปเป็นสัญญาณดิจิตอล เพื่อส่งให้กับ

เครื่องไมโครคอมพิวเตอร์

5) บัลลาสต์อิเล็กทรอนิกส์ ใช้ในการขับกำลังให้กับหลอดฟลูออเรสเซ็นต์ ซึ่งเป็นแหล่งจ่ายแสงที่มีแสงสม่ำเสมอตามความยาวของหลอด และการใช้บัลลาสต์ อิเล็กทรอนิกส์ที่ความถี่สูง แสงที่ได้จะมีการกระเพื่อมน้อย ทำให้มีผลดีต่อระบบการอ่านภาพ

จากระบบที่ได้ทำการทดลองอ่านภาพไปยังเครื่องไมโครคอมพิวเตอร์ก็ได้ประสพกับปัญหาหลายอย่าง เช่น Repeatability ของข้อมูลที่ได้จากการอ่านภาพไม่ค่อยดีนัก, ความเพี้ยนอันเกิดจากเลนส์, บริเวณขอบของภาพยังมีความชัดเจนไม่เพียงพอ และ แสงที่ให้กับแผ่นโลหะมีการกระเพื่อม

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 3.6 ระบบการอ่านภาพ แบบ ที่ใช้ในการทดลอง

3.6.2 ผลการทดลองและปัญหาที่พบ

1) ความเพี้ยนของเลนส์

เมื่อดูข้อมูลในตารางที่ 3.1(แถว REAL AVG) จะเห็นได้ว่าบริเวณ ปลายๆค่าจะน้อยกว่าบริเวณตรงกลาง ก็คือเกิดความเพี้ยนขึ้น (Distrotion) ซึ่งเป็นแบบถังน้ำ (Barrel) มีความเพี้ยนสูงสุด 1 มิลลิเมตร และต่ำสุด 0.1 มิลลิเมตร

POSITION	1	2	3	4	5	6	7	8	9	10	11	12	13
SAMPLE 1	3.78	9.48	3.84	9.78	3.94	9.78	3.94	9.84	3.92	9.78	3.86	9.58	3.82
SAMPLE 2	3.78	9.46	3.86	9.78	3.92	9.82	3.92	9.84	3.92	9.78	3.86	9.58	3.82
SAMPLE 3	3.78	9.46	3.88	9.68	3.92	9.88	3.94	9.84	3.92	9.78	3.86	9.58	3.82
SAMPLE 4	3.78	9.46	3.88	9.68	3.94	9.78	3.94	9.84	3.92	9.78	3.86	9.58	3.82
SAMPLE 5	3.78	9.46	3.88	9.68	3.92	9.88	3.94	9.84	3.92	9.78	3.86	9.58	3.82
SAMPLE 6	3.78	9.46	3.88	9.68	3.94	9.78	3.96	9.82	3.94	9.76	3.86	9.58	3.82
SAMPLE 7	3.78	9.46	3.88	9.68	3.94	9.78	3.96	9.82	3.94	9.76	3.86	9.58	3.82
SAMPLE 8	3.78	9.48	3.86	9.68	3.94	9.88	3.92	9.84	3.92	9.78	3.86	9.58	3.82
SAMPLE 9	3.78	9.46	3.88	9.68	3.94	9.78	3.96	9.82	3.94	9.76	3.88	9.58	3.82
SAMPLE 10	3.78	9.46	3.88	9.68	3.92	9.82	3.92	9.86	3.98	9.78	3.86	9.58	3.82
DATA MIN	3.78	9.46	3.84	9.68	3.92	9.78	3.92	9.82	3.98	9.76	3.86	9.58	3.82
DATA MAX	3.78	9.48	3.88	9.78	3.94	9.82	3.96	9.86	3.94	9.78	3.88	9.58	3.82
DIFF	0.00	0.02	0.04	0.02	0.02	0.04	0.04	0.04	0.04	0.02	0.02	0.00	0.00
REAL DIFF	0.00	0.00	0.01	0.00	0.01	0.01	0.01	0.01	0.01	0.00	0.01	0.00	0.00
AVERAGE	3.78	9.46	3.87	9.68	3.93	9.79	3.94	9.84	3.92	9.77	3.86	9.58	3.82
REAL AVG	0.95	2.37	0.97	2.42	0.98	2.45	0.99	2.46	0.98	2.44	0.97	2.40	0.96

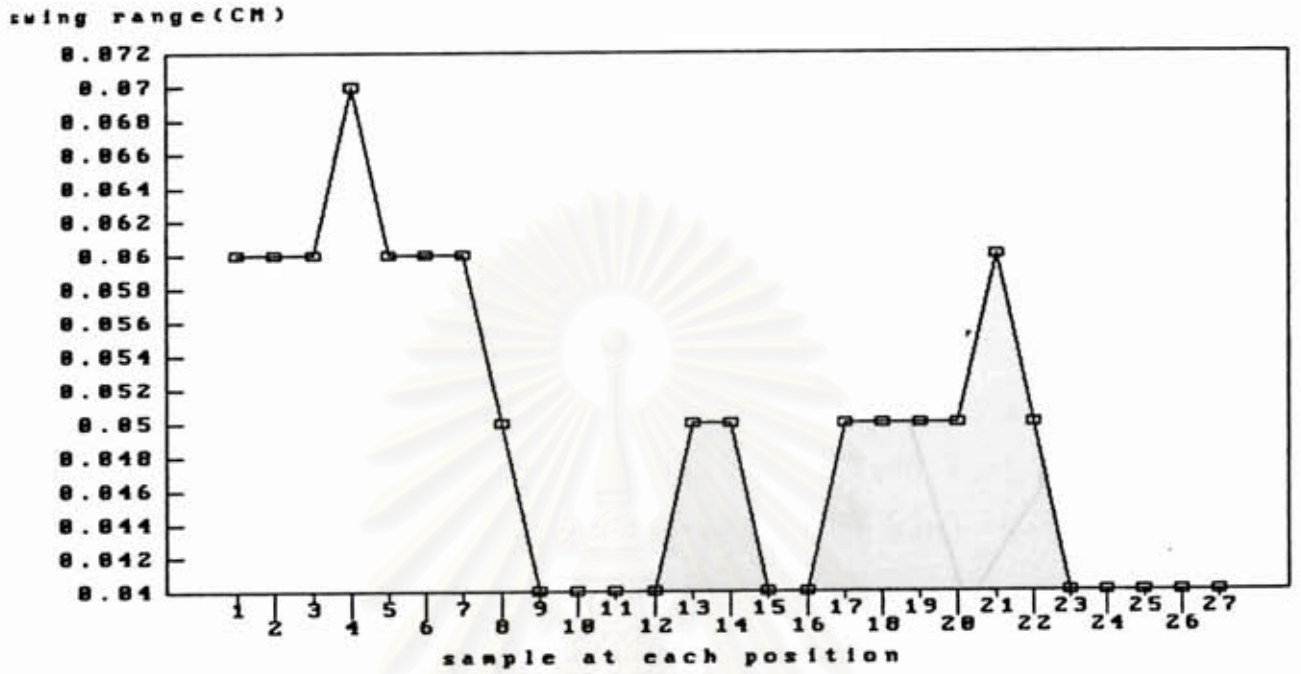
ตารางที่ 3.1 ค่าที่บันทึกได้จากการทดลองที่ระยะ 50 CM

2) ขอบของภาพไม่ชัดเจน

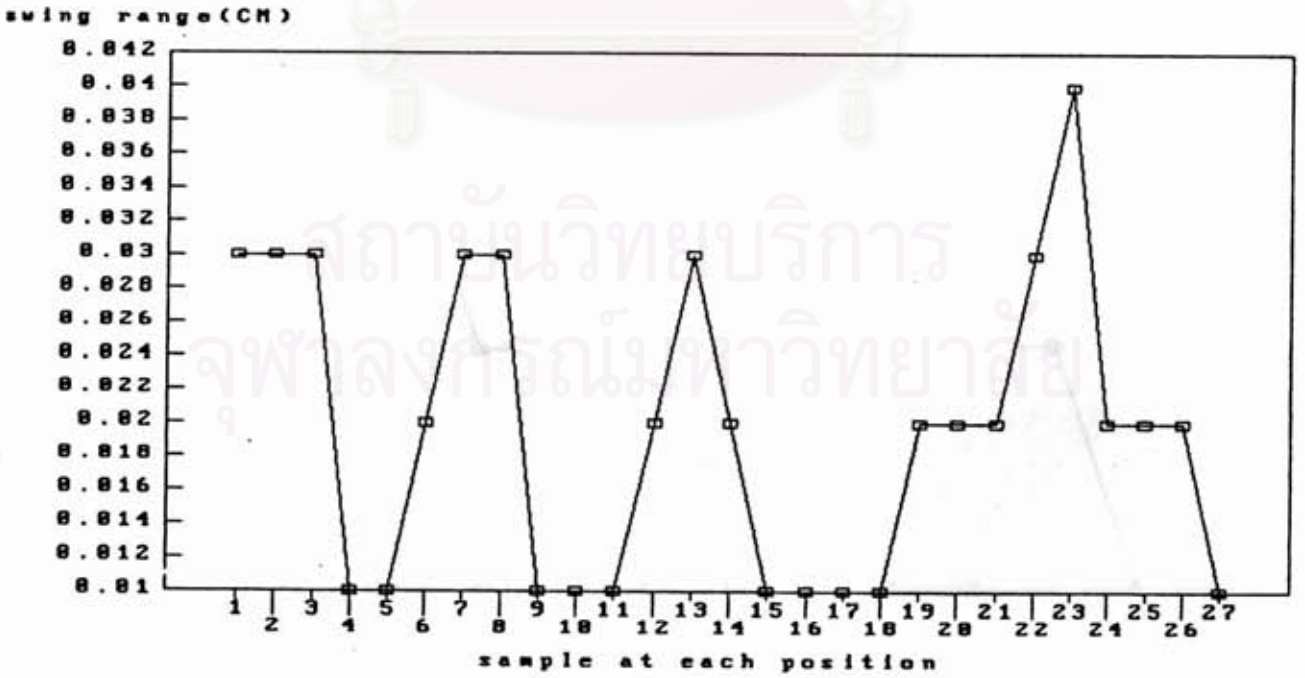
ความชัดเจนของภาพที่ผ่านเลนส์นั้น ในทางปฏิบัติบริเวณขอบของภาพจะมัว ทำให้ขอบของสัญญาณมีการแกว่งเช่นเดียวกัน ค่าที่ได้จากการอ่านก็เลยไม่แน่นอน ซึ่งขึ้นกับระยะห่างในการจับภาพ สังเกตุได้โดยการเปรียบเทียบระหว่าง รูปที่ 3.7a กับ รูปที่ 3.8a หรือ รูปที่ 3.7b กับ รูปที่ 3.8b จากกราฟสังเกตุได้ว่าที่ระยะห่าง 25 เซ็นติเมตร มีระยะการเปลี่ยนแปลงค่า(Swing Range)น้อยกว่า ที่ระยะห่าง 50 เซ็นติเมตร

3) แสงไม่สม่ำเสมอ และ มีการกระเพื่อม

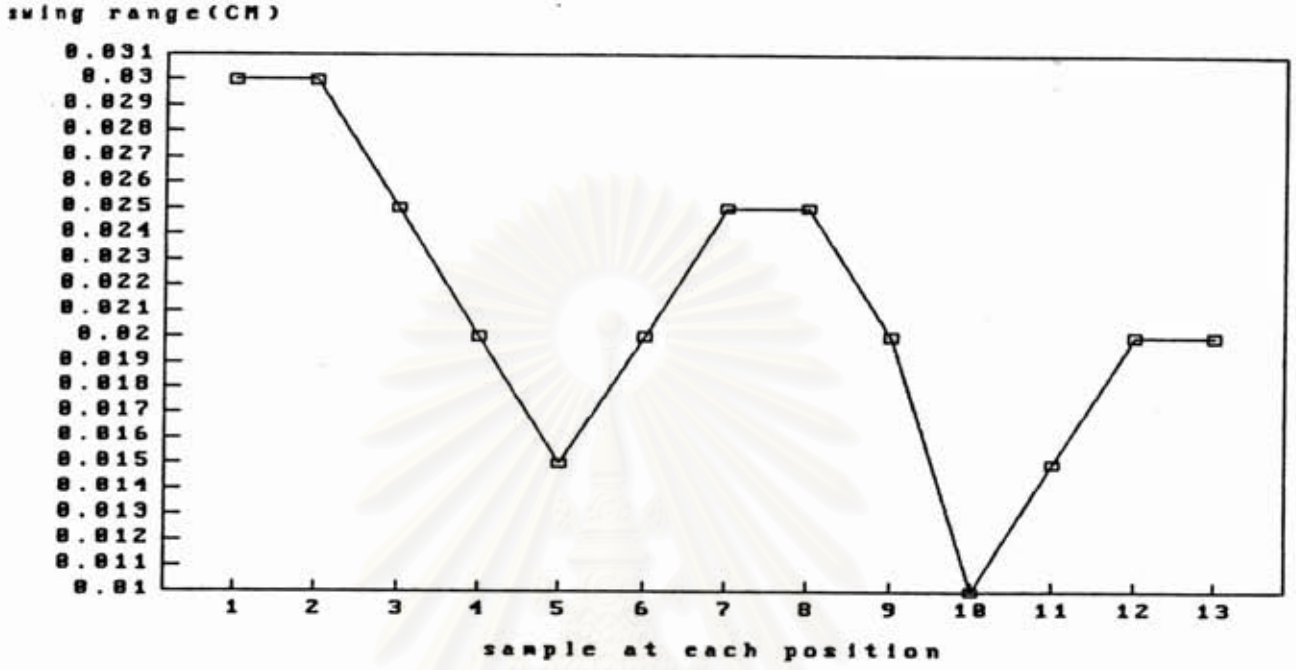
จากการทดลอง ถ้าใช้หลอดกลมแสงที่ได้ไม่สม่ำเสมอ และกรณีใช้หลอดไส้แบบยาวๆ กำลังไฟฟ้าที่ใช้สูงยากต่อการควบคุมกระแสไฟฟ้าให้เรียบ และถึงแม้จะใช้หลอดฟลูออเรสเซ็นต์ซึ่งมีแสงสม่ำตลอดความยาว แต่ก็มีปัญหาเรื่องแสงกระพริบ สังเกตุได้โดยการเปรียบเทียบระหว่าง รูปที่ 3.7a กับ รูปที่ 3.7b หรือ ระหว่าง รูปที่ 3.8a กับ รูปที่ 3.8b



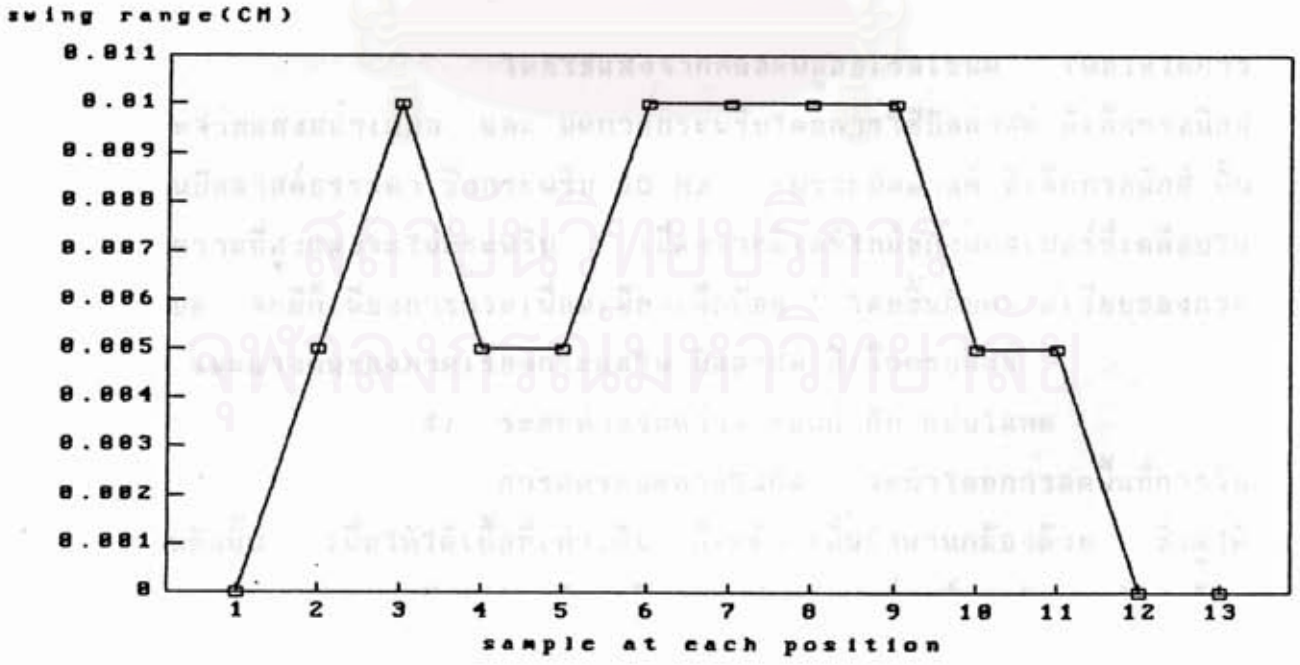
รูปที่ 3.7a ค่าที่ได้จากการทดลองที่ระยะห่าง 50 cm,แสงกระเพื่อมมาก



รูปที่ 3.7b ค่าที่ได้จากการทดลองที่ระยะห่าง 50 cm,แสงกระเพื่อมน้อย



รูปที่ 3.8a ค่าที่ได้จากการทดลองที่ระยะห่าง 25 cm,แสงกระเพื่อมมาก



รูปที่ 3.8b ค่าที่ได้จากการทดลองที่ระยะห่าง 25 cm,แสงกระเพื่อมน้อย

4) ระยะห่างระหว่าง เลนส์ กับ แผ่นโลหะ

จากการคำนวน การจับภาพที่มีเนื้อที่มากจะต้องใช้ระยะทางโฟกัสห่างมาก หรือถ้าจะใช้เลนส์มุมกว้าง ก็มีความเพี้ยนสูง ดังแสดงในรูปที่ 3.9 และการลดระยะทางโฟกัส โดยการใช้กระจกสะท้อนนั้นมีความยุ่งยากในทางปฏิบัติ

3.6.3 การแก้ไขปัญหา

1) ความเพี้ยนของเลนส์

ได้ทำการทดลองหาเลนส์ที่มีความเพี้ยนต่ำ ที่ระยะทางโฟกัสใกล้ และ สามารถหาได้ภายในประเทศ เมื่อพิจารณาแล้วเลนส์ชนิดไมโครมีความเหมาะสมกับการจับภาพใกล้ๆ ดังนั้นจึงได้เลือกเลนส์ไมโคร 55 mm F2.8 AI ของยี่ห้อ NIKON ที่มีมุมการรับภาพ 43 องศา แต่ในการใช้งานจะลดมาใช้ที่มุมรับภาพ 30 องศา

2) ขอบของภาพไม่ชัดเจน

การที่จะทำให้ขอบภาพชัดเจนขึ้นนั้น ทำได้โดยการนำเลนส์ไปใกล้แผ่นโลหะ เพื่อเป็นการลดพื้นที่การจับภาพ ดังนั้นจึงต้องเพิ่มจำนวนกล้อง และ ทำการเพิ่มปริมาณแสงโดยใช้หลอดฟลูออเรสเซ็นต์ 2 หลอด

3) แสงไม่สม่ำเสมอ และ มีการกระเพื่อม

โดยใช้แสงจากหลอดฟลูออเรสเซ็นต์ เพื่อให้ได้การกระจายแสงสม่ำเสมอ และ ลดการกระพริบโดยการใช้บัลลาสต์ อิเล็กทรอนิกส์แทนบัลลาสต์ธรรมดา ซึ่งกระพริบ 50 Hz เพราะบัลลาสต์ อิเล็กทรอนิกส์ นั้นใช้ความถี่สูงแสงจะไม่กระพริบ เนื่องจากดีเลย์ไทม์ของฟอสเฟอร์ที่เคลือบในหลอด จะมีก็เพียงการกระเพื่อมเพียงเล็กน้อย โดยขึ้นกับความเรียบของกระแส และแรงดันของภาคเรียงกระแสใน บัลลาสต์ อิเล็กทรอนิกส์

4) ระยะห่างระหว่าง เลนส์ กับ แผ่นโลหะ

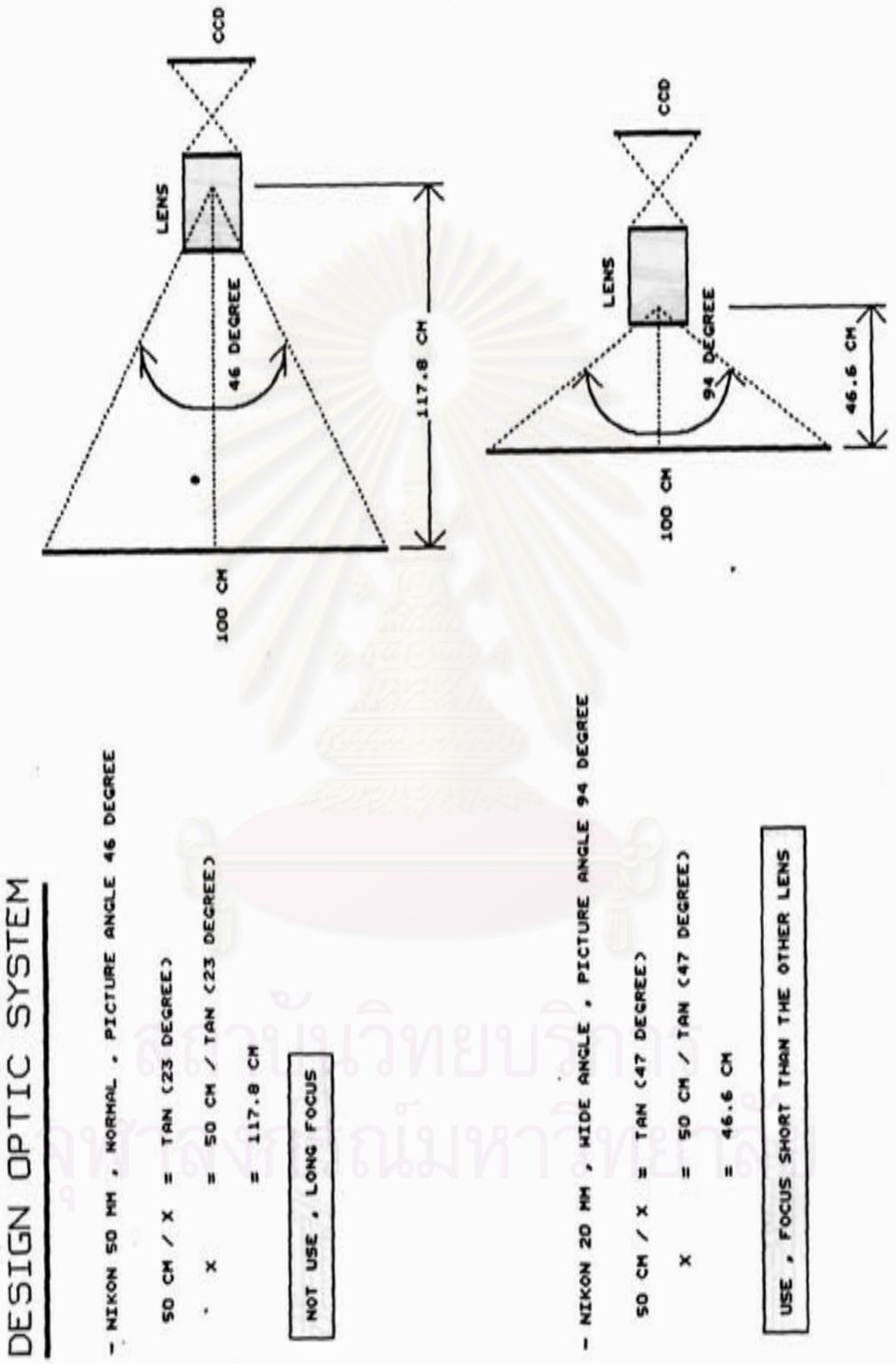
การลดระยะทางโฟกัส จะทำโดยการลดพื้นที่การจับภาพดังนั้น เพื่อให้ได้เนื้อที่เท่าเดิม ก็จะต้องเพิ่มจำนวนกล้องด้วย ซึ่งดูได้จากผลการคำนวณมุมรับภาพเปรียบเทียบระหว่าง เลนส์นอร์มอลกับเลนส์มุมกว้าง ใน รูปที่ 3.9 ทั้งนี้เพื่อแสดงให้เห็นว่าหากใช้เลนส์ที่มีมุมการรับภาพน้อยจะทำให้ระยะโฟกัสมากขึ้น ดังนั้นถ้าใช้เลนส์ที่มุมการรับภาพน้อยและลดระยะห่างให้น้อยลงจะต้องเพิ่มจำนวนกล้องเพื่อให้สามารถครอบคลุมเนื้อที่เท่าเดิม

เมื่อดูปัญหาที่เกิดขึ้นดังที่ได้กล่าวเอาไว้จึงควรเลือกระยะห่างระหว่างเลนส์กับแผ่นโลหะที่น้อยกว่า ซึ่งในที่นี้คือ 25 เซ็นติเมตร แต่อย่างไรก็ตามเนื่องจากมุมการรับภาพที่มากของเลนส์ไมโคร คือ 43 องศา ทำให้เกิดการมองเห็นความหนาของแผ่นโลหะขึ้น การลดปัญหานี้โดยการลดมาใช้ที่มุมรับภาพ 30 องศา ดังนั้นเพื่อการเพิ่มเนื้อที่การอ่านจึงต้องเพิ่มระยะทางระหว่างเลนส์กับแผ่นโลหะเป็น 46.6 เซ็นติเมตร โดยทำให้การรับภาพครอบคลุมความกว้าง 25 เซ็นติเมตรพอดี และจะต้องมีส่วนที่เหลื่อมกันสำหรับการต่อภาพ ซึ่งได้เผื่อปลายทั้ง 2 ข้าง ข้างละ 2.5 เซ็นติเมตร ทำให้ต้องใช้กล้อง 5 ชุด ดังรูปที่ 4.30

ในการเลือกทิศทางของแสง แม้ว่าการอ่านทวนแสงจะดีในแง่ความแน่นอนของข้อมูล แต่ผลข้างเคียงที่เกิดจากความหนานั้นเกิดขึ้นมากกว่าแบบใช้แสงสะท้อน ดังนั้นจึงเลือกอ่านภาพโดยใช้แสงสะท้อน ซึ่งสีของแผ่นโลหะจะต้องมีสีขาวตัดกับฉากสีดำ(ผิวโต๊ะ) เพื่อให้เครื่องแยกขอบเขตของแผ่นโลหะได้ชัดเจน และ การเพิ่มความชัดเจนยังได้ทำในส่วนของแสง โดยการเพิ่มความเข้มแสงด้วยการใช้หลอดฟลูออเรสเซ็นต์ 2 หลอด

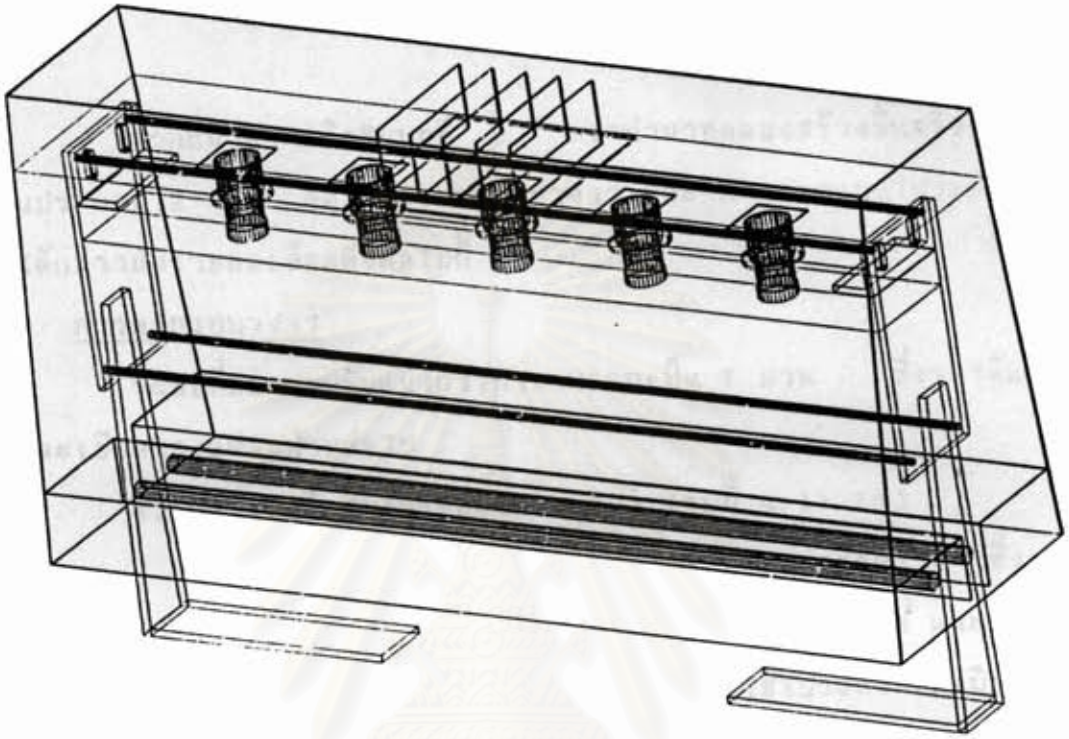
จากการทดลองทำให้เรารู้ขีดจำกัดและปัญหาต่างๆ ในการสร้างระบบการอ่านภาพ ซึ่งมีประโยชน์อย่างมากในการพิจารณาถึงความเป็นไปได้ แล้วนำมาแก้ไขปรับปรุงในการสร้างขึ้นจริง เพื่อให้ได้เครื่องอ่านภาพแผ่นโลหะเรียบที่มีคุณสมบัติตรงตามความต้องการของผู้ใช้

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

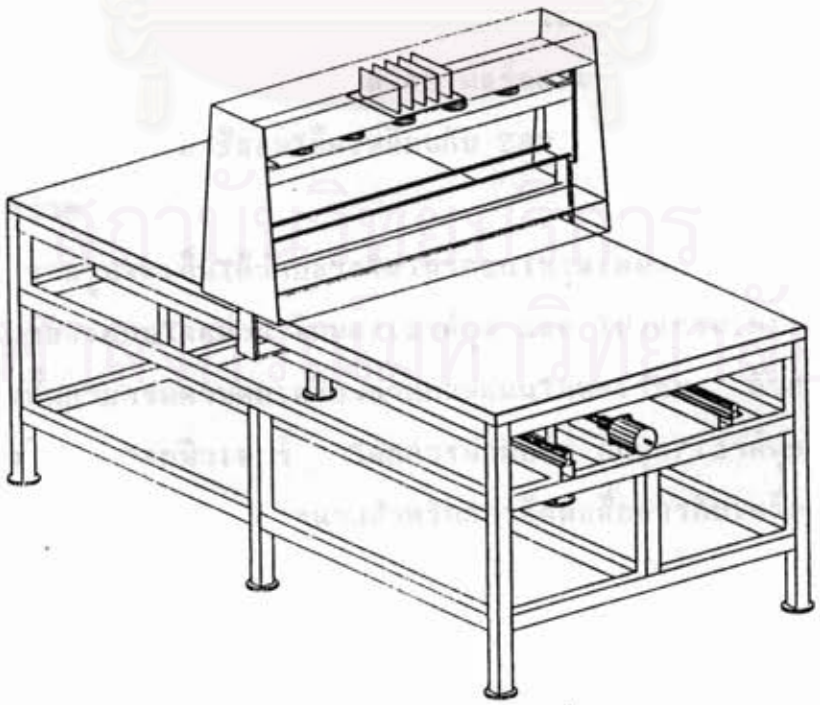


รูปที่ 3.9 ระยะทางโฟกัสที่ได้จากการคำนวณ เปรียบเทียบ
ระหว่าง เลนส์นอร์มอล กับ เลนส์มุมกว้าง

จากการทดลองที่ได้กล่าวไว้แล้ว ทำให้เราสามารถออกแบบลักษณะของเครื่องอ่านภาพแผ่นโลหะเรียบ ดังแสดงในรูปที่ 3.10 และ รูปที่ 3.11



รูปที่ 3.10 ลักษณะโครงสร้างส่วนหัวอ่านของเครื่องอ่านภาพแผ่นโลหะเรียบ



รูปที่ 3.11 ลักษณะโครงสร้างโดยรวมของเครื่องอ่านภาพแผ่นโลหะเรียบ

บทที่ 4

การสร้างเครื่องอ่านภาพแผ่นโลหะเรียบ

จากแนวความคิดในบทที่ 3 เราน่ามาทดลองสร้างขึ้นจริง โดยมีส่วนประกอบ 2 ส่วน คือ การออกแบบวงจร และ การออกแบบโปรแกรม ซึ่งจะได้กล่าวถึงรายละเอียดดังต่อไปนี้

4.1 การออกแบบวงจร

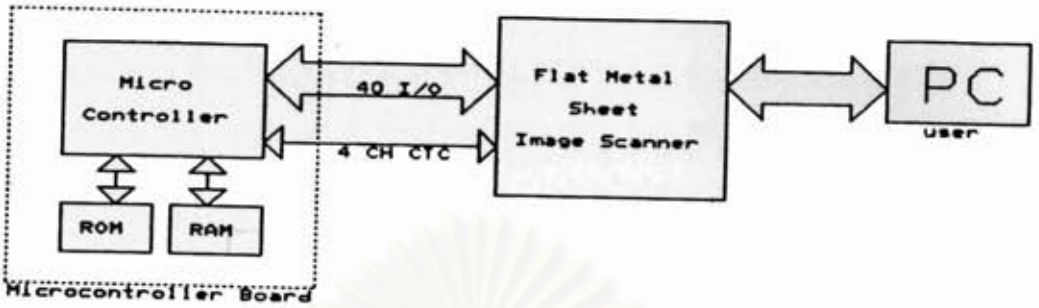
ระบบที่ออกแบบนั้นแบ่งการทำงานออกเป็น 7 ส่วน ซึ่งจะได้แสดงรายละเอียดของแต่ละส่วนต่อไป

4.1.1 บอร์ดไมโครคอนโทรลเลอร์ (รูปที่ 4.1) [16]

เป็นส่วนที่ใช้ควบคุมการทำงานของระบบฮาร์ดแวร์ ซึ่งเราสามารถซื้อหาได้โดยทั่วไป เพราะการอินเตอร์เฟสระหว่างบอร์ด และ การ์ดต่างๆนั้น ทำได้โดยผ่านพอร์ต อินพุต/เอาต์พุต และ ใช้โปรแกรมเมเบิลเคาน์เตอร์ในการสร้างไทม์มิ่งสำหรับควบคุม CCD อิมเมจเซ็นเซอร์

ดังนั้นเพื่อขนาดบอร์ดที่เล็ก และได้ประสิทธิภาพ เราจึงเลือกใช้ซีพียูประเภทที่เป็นไมโครคอนโทรลเลอร์ ในที่นี้เราใช้ Z84C11 ของบริษัท ZILOG ซึ่งภายในตัวนี้มีพอร์ต อินพุต/เอาต์พุต 40 พอร์ตและโปรแกรมเมเบิลเคาน์เตอร์ 4 ตัว ทั้งยังสามารถใช้ออพโค๊ดเหมือนกับ Z80 ทำให้สามารถหาซอฟต์แวร์สนับสนุนได้ง่าย

จากรูปจะเห็นได้ว่าบอร์ดไมโครคอนโทรลเลอร์ จะสามารถควบคุมการทำงานของระบบโดยผ่าน อินพุต/เอาต์พุต และ โปรแกรมเมเบิลเคาน์เตอร์ ซึ่งผู้ใช้เองนั้นสามารถควบคุมเครื่องอ่านภาพแผ่นโลหะเรียบ ด้วยการส่งคำสั่งผ่านเครื่องไมโครคอมพิวเตอร์ โดยการน่าพอร์ต อินพุต/เอาต์พุต ของบอร์ดนี้จำนวนหนึ่ง มาใช้เป็นพอร์ตขนานสำหรับการติดต่อสื่อสารกับเครื่องไมโครคอมพิวเตอร์



รูปที่ 4.1 บอร์ดไมโครคอนโทรลเลอร์

4.1.2 เมนบอร์ด (ดังแสดงในรูปที่ 4.2)

การออกแบบส่วนนี้ เพื่อทำให้ระบบนั้นมีลักษณะแบ่งออกเป็นโมดุล โดยจะใช้บัสร่วมกันในรูปแบบเป็นการ์ดเสียบ ซึ่งมีสัญญาณต่างๆที่ใช้ร่วมกันอยู่ เช่น

แอดเดรสบัส ใช้สำหรับการบอกตำแหน่งของจุดรับภาพภายในตัวอิมเมจเซ็นเซอร์ กรณีที่มีข้อมูลการเปลี่ยนแปลงเกิดขึ้นที่กล้องใดก็จะเก็บแอดเดรสในขณะนั้น ที่หน่วยความจำ FIFO ในการ์ดควบคุมกล้องที่เป็นของกล้องตัวนั้น

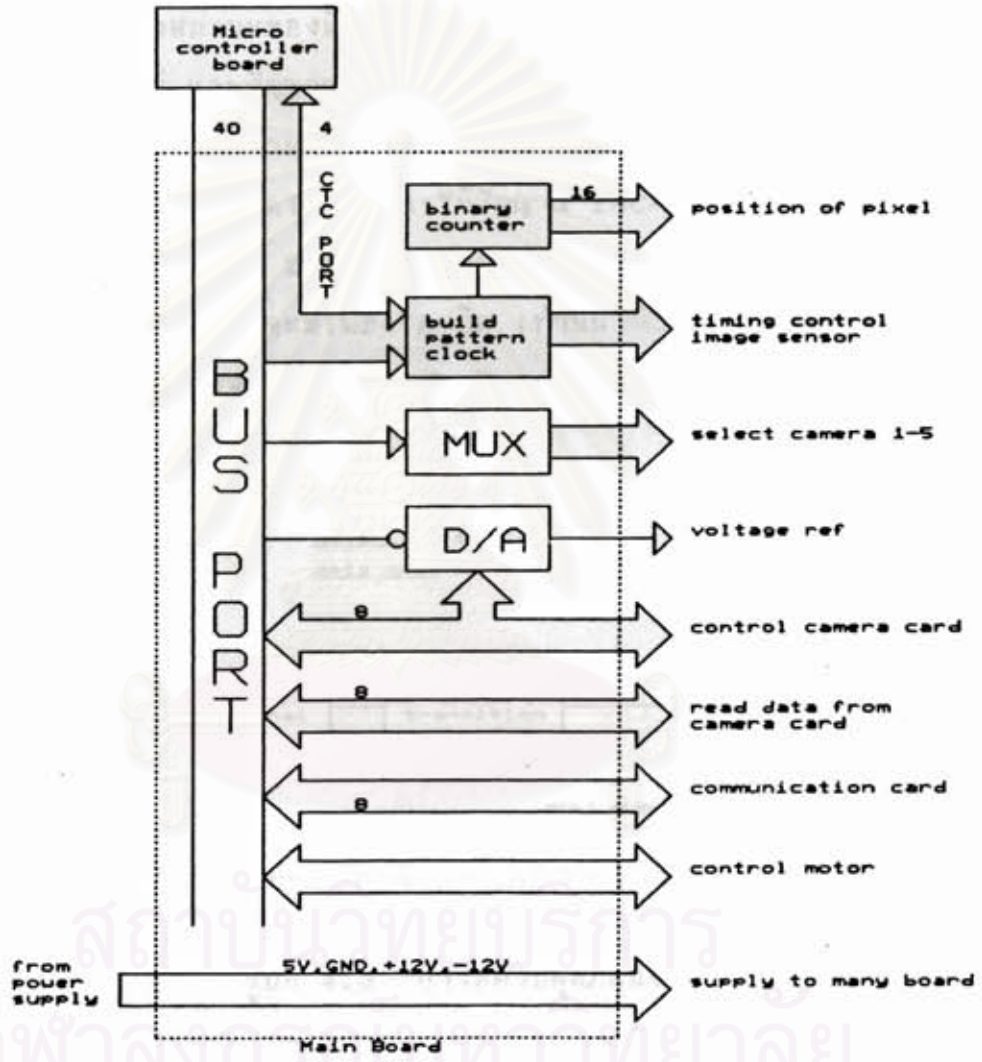
สัญญาณ ไทม์มิ่ง คอนโทรล อิมเมจเซ็นเซอร์ ทำหน้าที่ในการควบคุมการอ่านข้อมูลออกจากตัวอิมเมจเซ็นเซอร์ ซึ่งสัญญาณไทม์มิ่ง นี้สร้างขึ้นจากส่วนที่เรียกว่า โปรแกรมเมเบิลคล็อก โดยใช้โปรแกรมเมเบิลเคาน์เตอร์ภายในไมโครคอนโทรลเลอร์บอร์ดมาทำ

ดาต้าบัส ใช้สำหรับการอ่านข้อมูลแอดเดรสจากหน่วยความจำ FIFO ภายในการ์ดควบคุมกล้อง ซึ่งใช้ร่วมกับสัญญาณ cam 1 - cam 5 ในการเลือกว่าจะอ่านข้อมูข้อมูลจากการ์ดใด โดยใช้สัญญาณควบคุมการอ่านร่วมกัน

แรงดันอ้างอิง สร้างขึ้นโดย D/A converter จะใช้ร่วมกันทุกๆการ์ดสำหรับการเปรียบเทียบระดับความเข้มของสัญญาณภาพจากอิมเมจเซ็นเซอร์เพื่อเปลี่ยนข้อมูลให้มีระดับเพียง 2 ระดับ คือ "0" และ "1" การโปรแกรมแรงดันนั้นทำโดยบอร์ดไมโครคอนโทรลเลอร์

การจ่ายแรงดันเพื่อเลี้ยงบอร์ดต่างๆนั้น ทำโดยผ่านบัสเช่นกัน แรงดันที่ส่งไปยังบอร์ดต่างๆในระบบบัส มี 5V GND +12V และ -12V

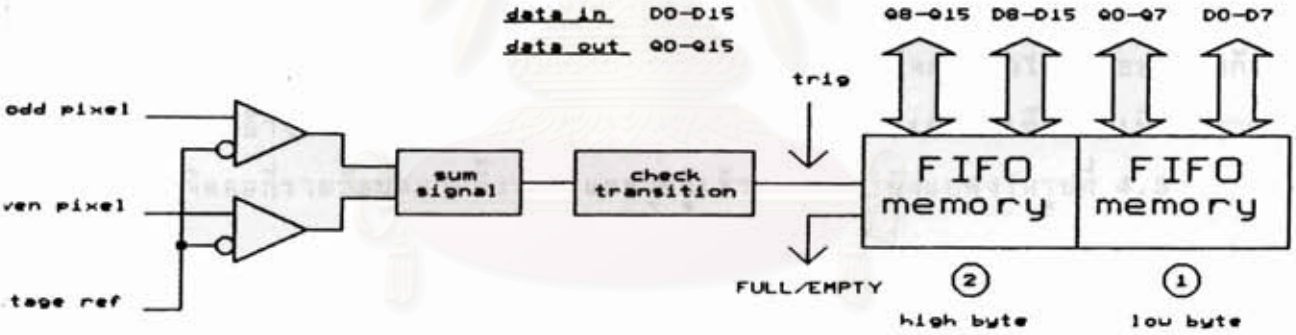
T149366



รูปที่ 4.2 เมนบอร์ด

4.1.3 การ์ดควบคุมกล้อง

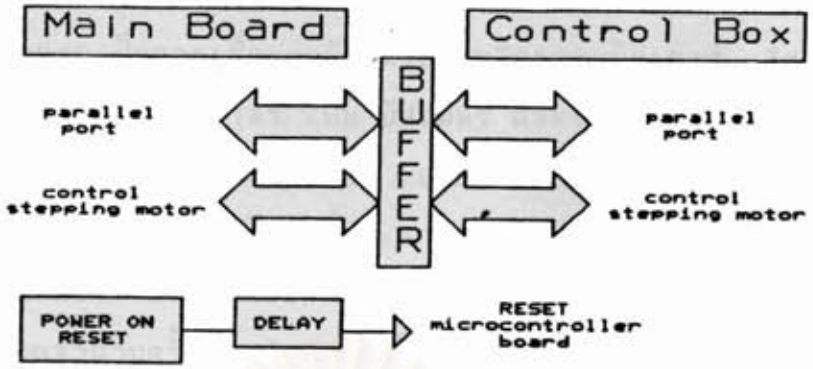
การ์ดนี้จะรับสัญญาณภาพ จากบอร์ดวงจรขยายสัญญาณภาพมาทำการเปรียบเทียบกับแรงดันอ้างอิงที่โปรแกรมไว้ เมื่อมีการเปลี่ยนแปลงระดับความเข้มของสัญญาณภาพ ก็จะได้สัญญาณการเปลี่ยนแปลงไปทริกหน่วยความจำ FIFO ให้เก็บข้อมูลแอดเดรสจุดภาพนั้นไว้ทางอินพุต D_0-D_7 โดยหน่วยความจำ FIFO มีสัญญาณบอกสถานะของมัน เช่น สัญญาณ /EMPTY บอกสถานะว่าหน่วยความจำว่างหรือไม่ และสัญญาณ /FULL บอกสถานะว่าหน่วยความจำเต็มหรือไม่ การอ่านข้อมูลจากหน่วยความจำ FIFO สามารถทำได้โดยอ่านทางเอาต์พุต Q_0-Q_7 และ อ่านข้อมูลถัดไปด้วยการให้สัญญาณ INC-RD วงจรที่ออกแบบนี้จะทำการเก็บข้อมูลครั้งละ 2 ไบท์ และการอ่านข้อมูลนั้นจะทำครั้งละไบท์เพราะว่าพอร์ต อินพุต/เอาต์พุต จะมีเพียง 8 บิต เท่านั้น ดังนั้นจึงต้องมีการมัลติเพล็กซ์เพื่อการอ่านครั้งละไบท์ ดังแสดงในรูปที่ 4.3



รูปที่ 4.3 การ์ดควบคุมกล้อง

4.1.4 การ์ดวงจรสื่อสารข้อมูล

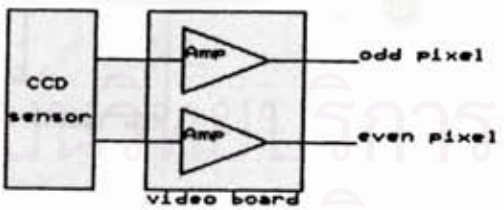
การ์ดนี้ไม่มีวงจรที่ซับซ้อน มีเพียงบัสไดร์เวอร์สำหรับทำหน้าที่เป็นบัฟเฟอร์ส่วนของการสื่อสารกับเครื่องไมโครคอมพิวเตอร์ สัญญาณการควบคุมการทำงานของสเต็บปิ้งมอเตอร์ และวงจรการรีเซ็ตระบบในตอนเปิดเครื่อง ทั้งนี้ก็เพื่อหน่วงเวลาให้อุปกรณ์ทุกๆส่วนพร้อมแล้ว จึงจะเริ่มการทำงานของระบบทั้งหมดได้ ดังแสดงในรูปที่ 4.4



รูปที่ 4.4 การ์ดวงจรสื่อสารข้อมูล

4.1.5 บอร์ดวงจรขยายสัญญาณภาพ

บอร์ดนี้จะติดตั้งกับส่วนเลนส์ และใช้ CCD ลิเนียร์อิมเมจเซ็นเซอร์ชนิดเลนส์ มีจำนวนจุดรับภาพ 5,000 จุด เป็นของ National/Panasonic รุ่น MN3664 [15] โดยจะถูกติดตั้งไว้บนบอร์ดนี้เพื่อขยายสัญญาณให้มีความแรงจนสามารถส่งสัญญาณไปได้ไกลมากขึ้น โดยไม่ถูกสัญญาณอื่นๆรบกวนได้ วงจรขยายสัญญาณของบอร์ดนี้ แบ่งเป็น 2 ส่วน เพื่อขยายจุดรับภาพของอิมเมจเซ็นเซอร์ทั้งจุดคี่ และจุดคู่ สัญญาณที่ได้จากบอร์ดนี้จะถูกนำไปเปรียบเทียบกับระดับอ้างอิงที่การ์ดควบคุมกล้อง สัญญาณที่ได้จากการเปรียบเทียบจะเป็นสัญญาณดิจิตอลที่รวมข้อมูลภาพทั้งจุดคี่ และจุดคู่เข้าด้วยกัน ดังแสดงในรูปที่ 4.5



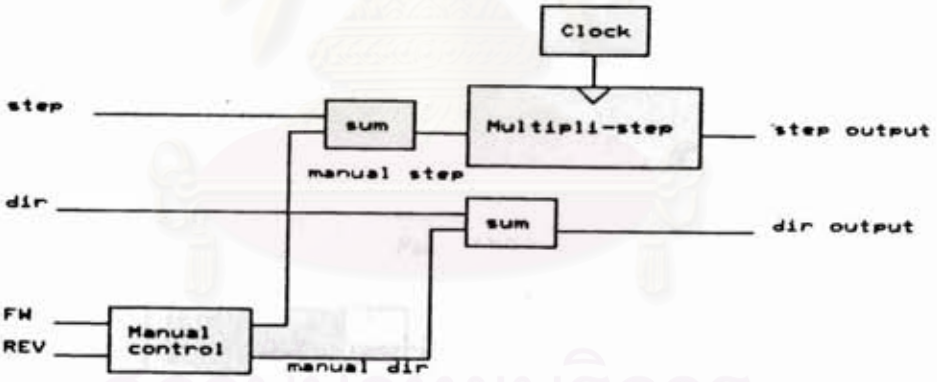
รูปที่ 4.5 บอร์ดวงจรขยายสัญญาณภาพ

4.1.6 บอร์ดอินเตอร์เฟส

บอร์ดนี้ไม่อยู่ภายในหัวอ่าน แต่จะแยกอยู่ในส่วนที่ไม่มีการเคลื่อนไหวเพื่อใช้เป็นจุดเชื่อมต่อกับผู้ใช้ ส่วนนั้น มีชื่อเรียกว่า "กล่องควบคุม" บอร์ดนี้แบ่งเป็น 2 ส่วน

ส่วนแรกเป็นการเชื่อมต่อกับเครื่องไมโครคอมพิวเตอร์ โดยผ่านทางบอร์ดขนาน ซึ่งก็มี บัสไดร์เวอร์ เป็น บัฟเฟอร์ และยังช่วยยกระดับของสัญญาณให้ดีขึ้น ส่วนนี้ไม่ซับซ้อนแต่ก็สำคัญมาก เพื่อที่จะสามารถส่งสัญญาณได้ไกลมากยิ่งขึ้น โครงสร้างของส่วนนี้ให้ดูรูปที่ 4.4

ส่วนที่สอง เป็นส่วนที่ใช้สำหรับควบคุมการทำงานของสเต็บปิ้งมอเตอร์ ซึ่งสามารถทวีคูณจำนวนครั้งของสัญญาณสเต็บจากที่บอร์ดไมโครคอนโทรลเลอร์ส่งมา เพื่อลดภาระการทำงานของซีพียู การควบคุมนั้นสามารถทำได้ 2 โหมด คือ แบบวงรอบเปิด(open loop) และ แบบวงรอบปิด(close loop) โดยการต่อตัวเอ็นโค๊ดเดอร์เพิ่มเข้าไป ซึ่งเราเลือกได้โดยจัมเปอร์ และการตั้งจำนวนการทวีคูณของสเต็บนั้นเลือกได้โดยการตั้งดิพสวิตช์ ในส่วนนี้ยังได้เพิ่มเอาการควบคุมด้วยมือ(Manual Control) เพื่อให้สามารถเลื่อนหัวอ่านไปยังตำแหน่งที่ต้องการโดยกดปุ่ม L direction และ ปุ่ม R direction สัญญาณสเต็บที่ได้จากบอร์ดนี้จะส่งไปยังอินพุตของสเต็บปิ้งมอเตอร์คอนโทรลเลอร์ เพื่อสร้างสัญญาณขับให้กับตัวสเต็บปิ้งมอเตอร์ต่อไป ดังแสดงในรูปที่ 4.6

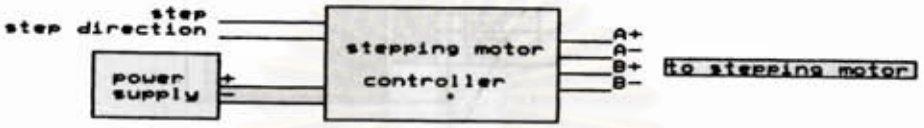


รูปที่ 4.6 บอร์ดอินเตอร์เฟส

4.1.7 สเต็บปิ้งมอเตอร์คอนโทรลเลอร์ และ การต่อใช้งาน [14]

ในส่วนนี้เป็นส่วนที่ใช้สำหรับสร้างสัญญาณขับสเต็บปิ้งมอเตอร์เพื่อเลื่อนตำแหน่งหัวอ่านภาพ โดยบอร์ดอินเตอร์เฟสจะส่งสัญญาณสเต็บไปให้สเต็บมอเตอร์คอนโทรลเลอร์ เพื่อให้มอเตอร์หมุน 1 สเต็บ โดยสัญญาณที่ขับสเต็บปิ้งมอเตอร์นั้นมี 2 เฟส และสามารถควบคุมทิศทางการหมุนโดยส่งสัญญาณบอกทิศทางให้กับสเต็บปิ้งมอเตอร์คอนโทรลเลอร์ ดังแสดงในรูปที่ 4.7

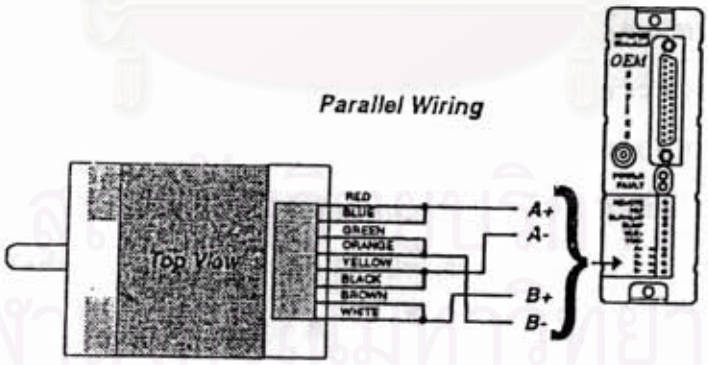
สัญญาณที่ส่งให้กับสเต็บปิ้งมอเตอร์ มี 2 เฟส โดยมีสายสัญญาณ 4 เส้น คือ A+,A-,B+ และ B- ซึ่งสัญญาณนี้สามารถใช้ขับสเต็บปิ้งมอเตอร์ได้โดยตรง เพราะภายในของสเต็บปิ้งมอเตอร์คอนโทรลเลอร์นั้นจะมีวงจรขยายสัญญาณให้มีกำลังขับอยู่แล้ว



รูปที่ 4.7 การต่อใช้งานสเต็บปิ้งมอเตอร์คอนโทรลเลอร์

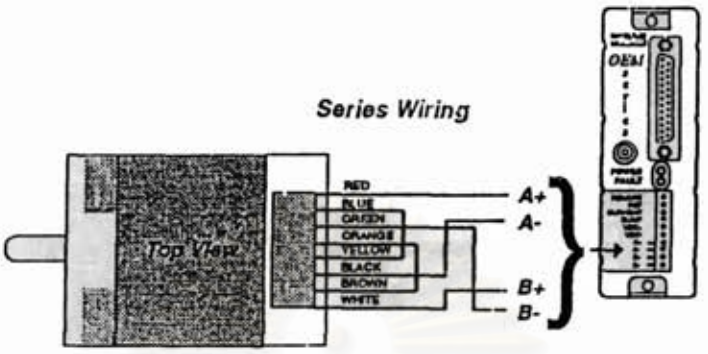
วงจรการต่อสเต็บปิ้งมอเตอร์นั้นมีอยู่ 2 แบบ คือ

การต่อแบบขนาน แบบนี้เหมาะสำหรับการใช้งานที่ต้องการกำลังบิดสูงๆ แต่ก็มีข้อเสียตรงที่ความเร็วของการหมุนค่อนข้างต่ำ วงจรการต่อในแบบนี้ดังแสดงในรูปที่ 4.8



รูปที่ 4.8 การต่อแบบขนาน

การต่อแบบอนุกรม แบบนี้เหมาะสำหรับการใช้งานที่ต้องการความเร็วสูงในการหมุน แต่ก็มีข้อเสียตรงที่แรงบิดที่ได้จะต่ำ วงจรการต่อในแบบนี้ดังแสดงในรูปที่ 4.9



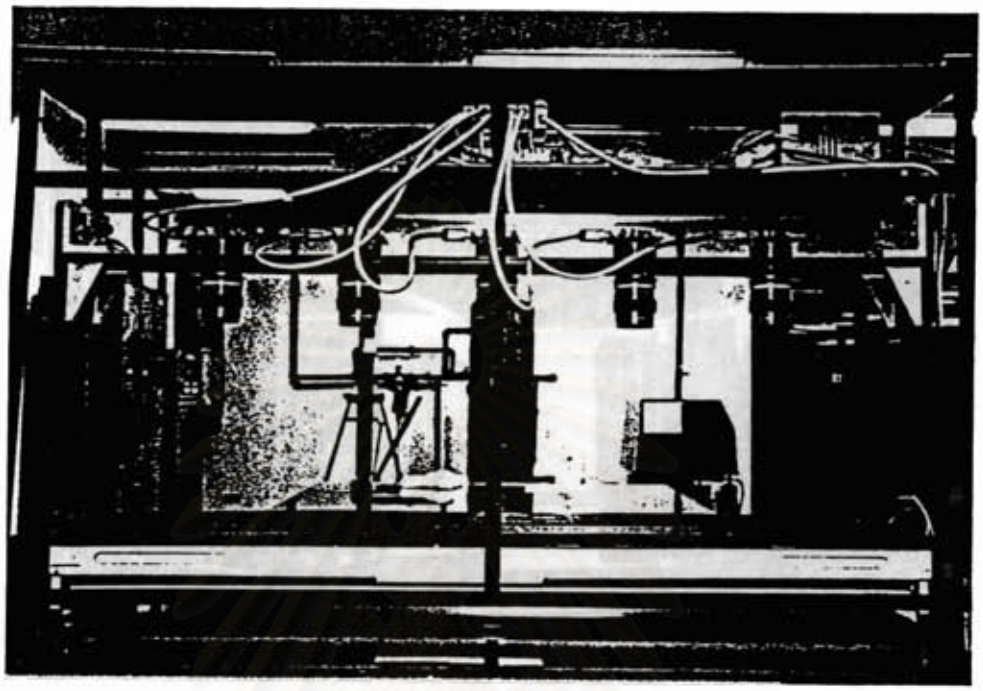
รูปที่ 4.9 การต่อแบบอนุกรม

จากแบบทั้งสองแบบนี้เราจะเลือกต่อแบบขนาน เพราะการออกแบบพยายามที่จะให้ระบบทำงานเร็ว ดังนั้นการต่อแบบนี้จึงต้องออกแบบให้หัวอ่านซึ่งเป็นโหลดของมอเตอร์นั้นมีน้ำหนักเบาที่สุดเท่าที่จะทำได้ แต่ก็ต้องคำนึงถึงความแข็งแรงด้วย เพื่อป้องกันการสั่นไหวในเวลาที่หัวอ่านเคลื่อนที่

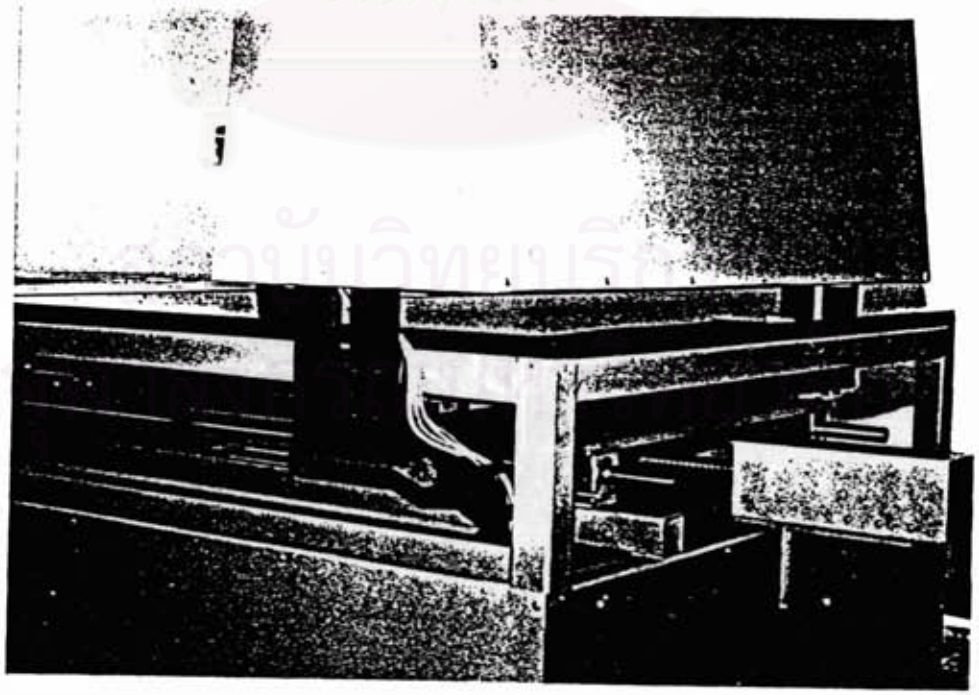
ส่วนประกอบของวงจรทั้งหมดของเครื่องอ่านภาพแผ่นโลหะเรียบที่ออกแบบขึ้นนี้ แยกออกเป็น 2 ส่วน

ส่วนแรก เป็นส่วนที่อยู่ในหัวอ่านซึ่งมีการเคลื่อนไหว ดังรูปที่ 4.10 ภายในจะมี บอร์ดไมโครคอนโทรลเลอร์, เมนบอร์ด, การ์ดวงจรควบคุมกล้อง, การ์ดวงจรการสื่อสารข้อมูล, บอร์ดวงจรขยายสัญญาณภาพ, CCD อิมเมจเซ็นเซอร์ และ สวิตชิ่งเพาเวอร์ซัพพลาย จากส่วนนี้จากมีสายเชื่อมต่อ โดยร้อยสายผ่านตีนตะขาบไปยังส่วนที่สอง ดังแสดงในรูปที่ 4.11

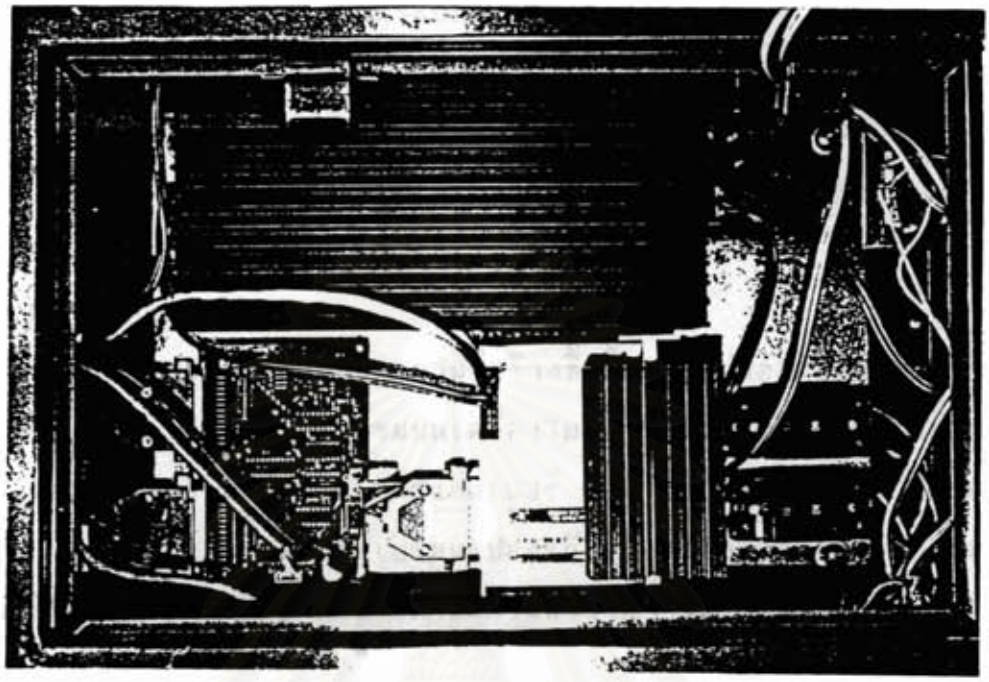
ส่วนที่สองเป็นกล่องควบคุม ภายในมี บอร์ดอินเตอร์เฟส, สเต็ปปิ้งมอเตอร์คอนโทรลเลอร์, สวิตชิ่งเพาเวอร์ซัพพลาย, เบรกเกอร์ และ พัดลม ส่วนภายนอกของคอนโทรลบอร์ดนั้นก็จะมี ฟิวส์, อินพุตไฟ 220 V, DB 25 ของพอร์ตขนาน และสวิตช์สำหรับการควบคุมระบบโดยผู้ใช้ คือ เพาเวอร์สวิตช์สำหรับการเปิดและปิดเครื่อง, สวิตช์ L direction และ สวิตช์ R direction ที่ใช้เมื่อต้องการเลื่อนหัวอ่าน รวมทั้ง พอร์ตขนานที่ใช้สำหรับการควบคุมการทำงานโดยเครื่องไมโครคอมพิวเตอร์ ดังแสดงในรูปที่ 4.12



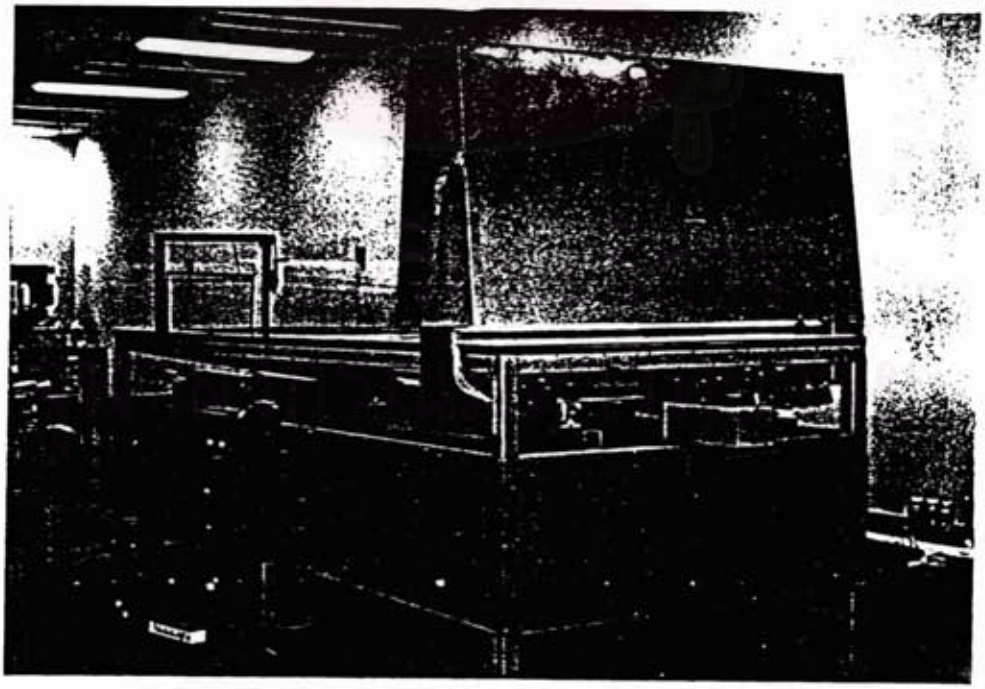
รูปที่ 4.10 เป็นส่วนหัวอ่านของเครื่องอ่านภาพแผ่นโลหะเรียบ



รูปที่ 4.11 แสดงการร้อยสายจากหัวอ่านผ่านตีนตะขาบมายังกล่องควบคุม



รูปที่ 4.12 เป็นส่วนที่อยู่ภายในของกล่องควบคุม



รูปที่ 4.13 ส่วนประกอบทั้งหมดของเครื่องอ่านภาพแผ่นโลหะเรียบ

ระบบทั้งหมดจะสามารถทำงานได้ ก็ต้องมีเครื่องไมโครคอมพิวเตอร์เป็นตัวควบคุมระบบ การเริ่มต้นระบบจะต้องเปิดเครื่องอ่านภาพแผ่นโลหะเรียบก่อน เพราะตัวมันถูกกำหนดให้เป็นตัวรับคำสั่ง ส่วนเครื่องไมโครคอมพิวเตอร์นั้นถูกกำหนดให้เป็นตัวส่งคำสั่ง เพื่อควบคุมการทำงานเครื่องอ่านภาพแผ่นโลหะเรียบ โดยภาพเครื่องทั้งหมดแสดงในรูปที่ 4.13 และ บล็อคของระบบทั้งหมดแสดงในรูปที่ 4.14

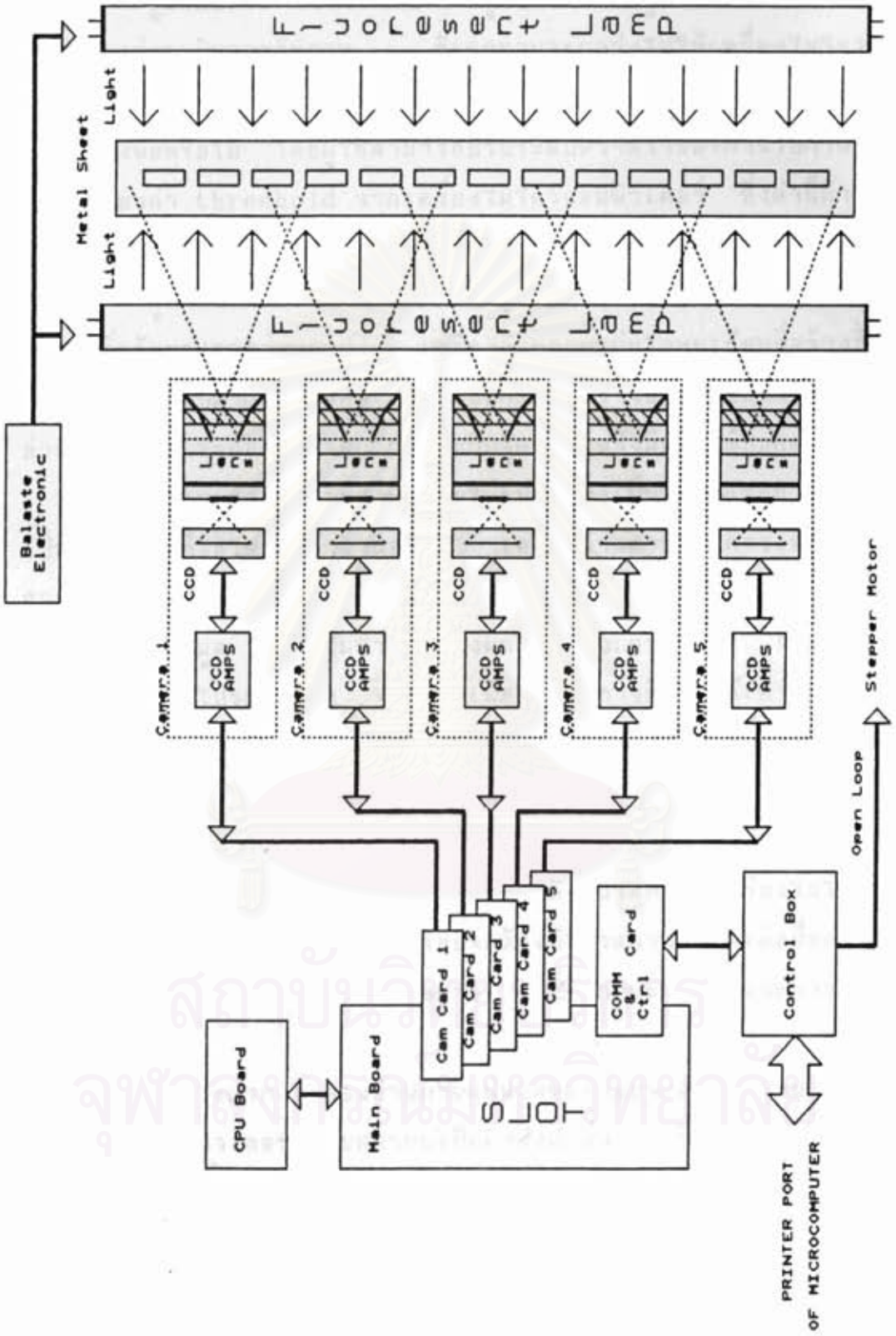
4.2 การออกแบบโปรแกรม

โปรแกรมระบบการอ่านภาพที่สร้างขึ้นนี้ ได้แบ่งออกเป็น 2 ส่วน

ส่วนแรก เป็นโปรแกรมบนเครื่องไมโครคอมพิวเตอร์ มีหน้าที่ในการส่งคำสั่งควบคุมไปยังเครื่องอิมเมจสแกนเนอร์ และ รับข้อมูลมาเก็บไว้ในคอมพิวเตอร์ เพื่อรวบรวมข้อมูลและเปลี่ยนแปลงแก้ไขข้อมูลให้ถูกต้องออกมา และยังทำหน้าที่ในการแสดงผลข้อมูลและสถานะการทำงาน ในรูปแบบต่างๆทั้งตัวอักษรและรูปกราฟฟิก รวมทั้งเป็นสื่อกลางที่ผู้ใช้สามารถควบคุมการทำงานของระบบอีกด้วย

อีกส่วน เป็นโปรแกรมบนเครื่องอ่านภาพแผ่นโลหะเรียบ ทำหน้าที่ควบคุมการทำงานฮาร์ดแวร์ของเครื่อง โดยรับคำสั่งจากเครื่องไมโครคอมพิวเตอร์มาปฏิบัติการ และส่งผลกลับใน 3 รูปแบบ คือ

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 4.14 บล็อคของระบบทั้งหมด

รูปแบบแรก จะส่งผลกลับในรูปแบบของข้อมูล เป็นโปรแกรมเพื่อการปรับแต่งระบบการรับภาพ ข้อมูลภาพจะถูกส่งไปให้เครื่องไมโครคอมพิวเตอร์แสดงผลเพื่อให้ผู้ใช้ทำการปรับแต่งได้อย่างถูกต้อง เช่น การปรับเลนส์และแสงเพียงพอหรือไม่ โดยผู้ใช้สามารถปรับระดับความไวของการรับภาพได้ โดยการเปลี่ยนค่า threshold จากเครื่องไมโครคอมพิวเตอร์ ซึ่งค่านี้ถ้าน้อยเกินไปก็จะมีสัญญาณรบกวนมาก ถ้ามากเกินไปก็จะไม่สามารถรับสัญญาณภาพนั้นไว้ได้

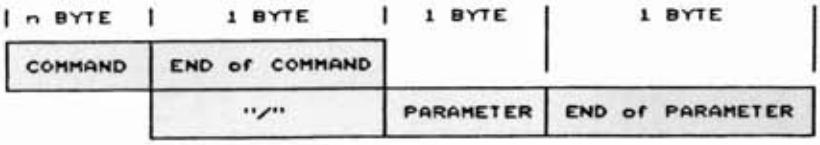
รูปแบบที่สอง จะส่งผลกลับในรูปแบบการเคลื่อนไหว เป็นโปรแกรมหาจุดเริ่มต้นของการอ่านภาพ เครื่องอ่านภาพแผ่นโลหะเรียบที่สร้างขึ้นนี้จะมีอุปกรณ์ตรวจสอบตำแหน่งติดอยู่(ในการออกแบบนี้เราใช้ไมโครสวิตช์) ส่วนหัวอ่านของเครื่องจะถอยหลังจนกว่าจะพบกับจุดเริ่มต้นซึ่งมีอุปกรณ์บอกตำแหน่งติดไว้

รูปแบบที่สาม จะส่งผลกลับในรูปแบบทั้งเป็นข้อมูลและการเคลื่อนไหว เป็นโปรแกรมที่ใช้ในการทำงานจริง เครื่องอ่านภาพจะมีการเคลื่อนไหวเพื่อกวาดอ่านตามระยะทางที่เรากำหนด และ ข้อมูลจะถูกส่งไปให้เครื่องไมโครคอมพิวเตอร์เก็บข้อมูล แต่จะไม่นำมาแสดงผลในรูปแบบกราฟฟิก เพราะจะทำให้การทำงานของโปรแกรมช้า จะบอกก็เพียงระยะทางที่อ่านได้เท่านั้น

4.2.1 โปรแกรมการติดต่อสื่อสาร เพื่อเชื่อมต่อการทำงานระหว่างเครื่องอ่านภาพแผ่นโลหะเรียบกับเครื่องไมโครคอมพิวเตอร์

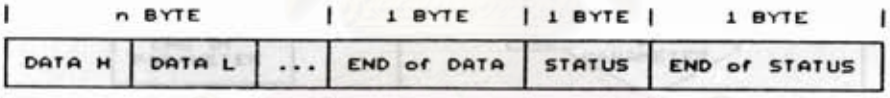
ในระบบของเครื่องอ่านภาพแผ่นโลหะเรียบนี้ ผู้ใช้สามารถควบคุมการทำงานโดยผ่านเครื่องไมโครคอมพิวเตอร์ ดังนั้นระหว่างเครื่องไมโครคอมพิวเตอร์กับเครื่องอ่านภาพแผ่นโลหะเรียบจะต้องมีโปรแกรมการติดต่อสื่อสาร เพื่อเชื่อมต่อให้ทำงานร่วมกันได้ตามต้องการ ซึ่งสถานะการรับและการส่งมีอยู่ 2 สภาวะ

สภาวะแรก เครื่องอ่านภาพแผ่นโลหะเรียบจะอยู่ในสถานะการรับคำสั่ง และเครื่องคอมพิวเตอร์อยู่ในสถานะเป็นตัวส่งคำสั่ง ซึ่งคำสั่งแยกออกเป็น 2 อย่าง คือ คำสั่งที่เป็นตัวกำหนดหน้าที่การทำงาน และ พารามิเตอร์เป็นตัวบอกขอบเขตการทำงาน โดยจะมีรูปแบบที่ถูกกำหนดไว้ดังแสดงในรูปที่ 4.15



รูปที่ 4.15 รูปแบบของคำสั่ง

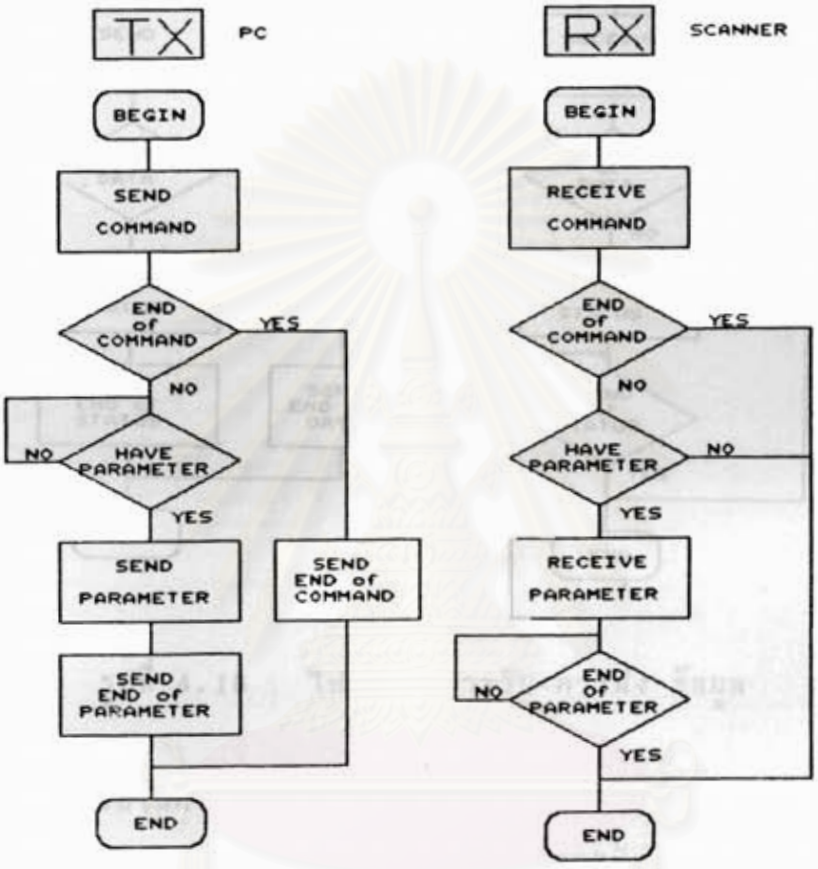
อีกสภาวะ เครื่องอ่านภาพแผ่นโลหะเรียบจะอยู่ในสถานะการส่งข้อมูล และเครื่องคอมพิวเตอร์อยู่ในสถานะเป็นตัวรับข้อมูล ซึ่งข้อมูลแยกออกเป็น 2 อย่าง คือ ข้อมูลที่ได้จากการสแกนและสถานะของข้อมูล ข้อมูลที่ได้นั้นเป็นแบบ 16 บิท ไบท์แรกเป็นไบท์สูง ไบท์ที่สองเป็นไบท์ต่ำสลับกันไป ทั้งนี้โดยสงวนข้อมูล OFFh ไว้สำหรับการตรวจสอบการจบข้อมูล และ การจบสถานะของข้อมูล โดยจะมีรูปแบบที่ถูกกำหนดไว้ดังแสดงในรูปที่ 4.16



รูปที่ 4.16 รูปแบบของข้อมูล

4.2.1.1 โปรแกรมการรับ-ส่งคำสั่ง

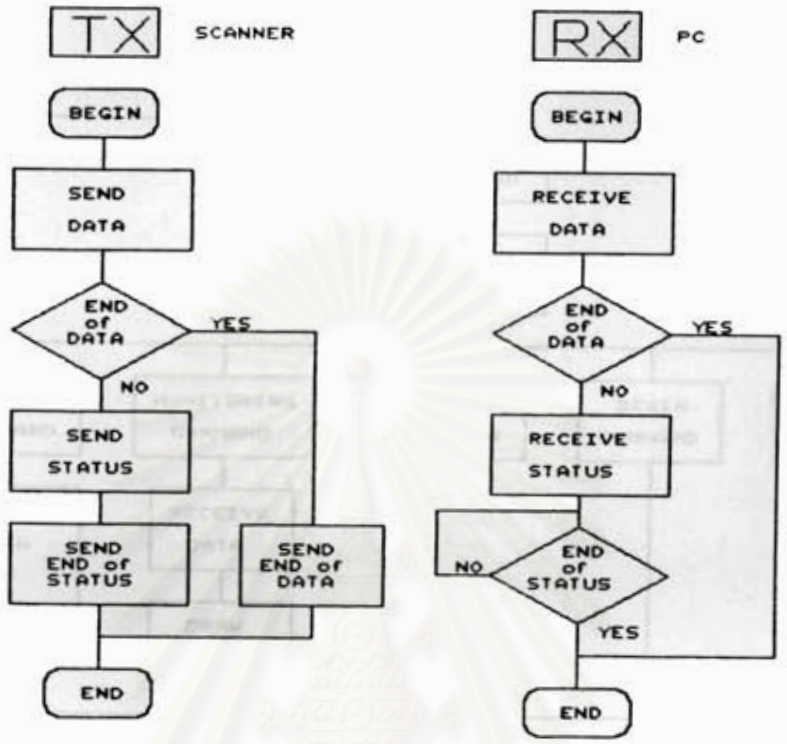
การทำงานของโปรแกรมบนเครื่องไมโครคอมพิวเตอร์จะทำการส่งคำสั่ง ส่วนเครื่องอ่านภาพแผ่นโลหะเรียบ จะทำหน้าที่การรับคำสั่งมาทำการปฏิบัติ โดยถ้ามีพารามิเตอร์จะต้องแทรกระหัส "/" เพื่อให้เครื่องอ่านภาพรู้ว่ามีพารามิเตอร์ตามมาด้วย ดังรูปที่ 4.17



รูปที่ 4.17 โปรแกรมการรับ-การส่ง คำสั่ง

4.2.1.2 โปรแกรมการรับ-ส่งข้อมูล

การทำงานของโปรแกรมบนเครื่องอ่านภาพแผ่นโลหะโดยทำการส่งข้อมูล ส่วนเครื่องไมโครคอมพิวเตอร์จะทำการรับข้อมูล โดยจะเป็นข้อมูล ไบท์สูงและไบท์ต่ำสลับกันไป จนกระทั่งพบระหัสการจบข้อมูล ซึ่งการจบข้อมูลจะตรวจสอบไบท์สูงหากพบระหัส OFFh จะเป็นจุดสิ้นสุดของข้อมูล และตามด้วยสถานะของข้อมูล ดังรูปที่ 4.18



รูปที่ 4.18 โปรแกรมการรับ-การส่ง ข้อมูล

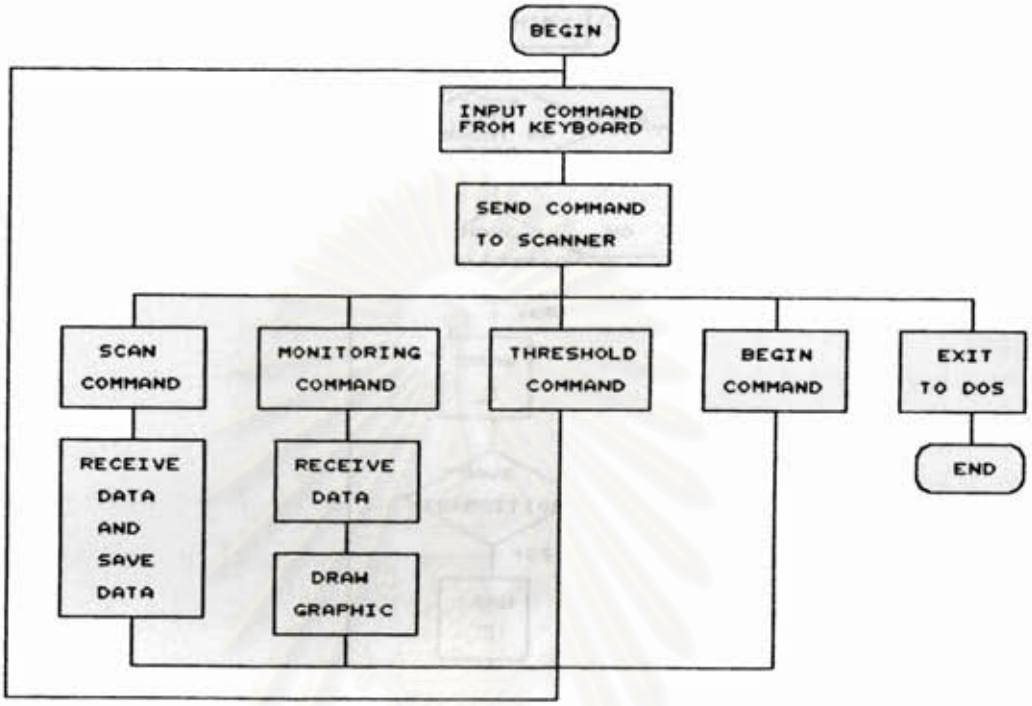
4.2.2 โปรแกรมในส่วนของเครื่องไมโครคอมพิวเตอร์

หน้าที่ของเครื่องไมโครคอมพิวเตอร์จะถูกใช้เป็นส่วนการติดต่อกับผู้ใช้ การเก็บข้อมูล และการแสดงผล ดังนั้นจึงต้องมีโปรแกรมในการควบคุมขบวนการเหล่านี้ ซึ่งจะได้อธิบายส่วนต่างๆต่อไปโดยลำดับ

4.2.2.1 โครงสร้างของโปรแกรมในเครื่องไมโครคอมพิวเตอร์

จากรูปที่ 4.19 ในส่วนแรก ๆ คือ การรับคำสั่งทางคีย์บอร์ดจากผู้ใช้ และส่งคำสั่งนั้นให้กับเครื่องอิมเมจสแกนเนอร์ ซึ่งก็ได้กล่าวมาแล้ว และหลังจากนั้นจะทำงานตามคำสั่งที่ได้เลือก โดยมี 5 คำสั่ง คือ SCAN, MONITORING, THRESHOLD, BEGIN และ EXIT คำส่ง THRESHOLD, BEGIN และ EXIT นั้นจะเป็นการส่งคำสั่งที่ไม่ต้องการการตอบกลับ ส่วนคำสั่ง SCAN นั้นหลังจากส่งคำสั่งแล้ว จะทำการรอรับข้อมูลจากเครื่องอ่านภาพมาเก็บไว้ในฮาร์ดดิสก์ โดยโครงสร้างของโปรแกรมทั้งหมด ดังแสดงในรูปที่ 4.19 และโครงสร้างของข้อมูลที่ได้จากการสแกน ดังแสดงในรูปที่ 4.20 ส่วนคำสั่ง

MONITORING นั้นมีรายละเอียดดังจะได้กล่าวต่อไปหลังจากนี้

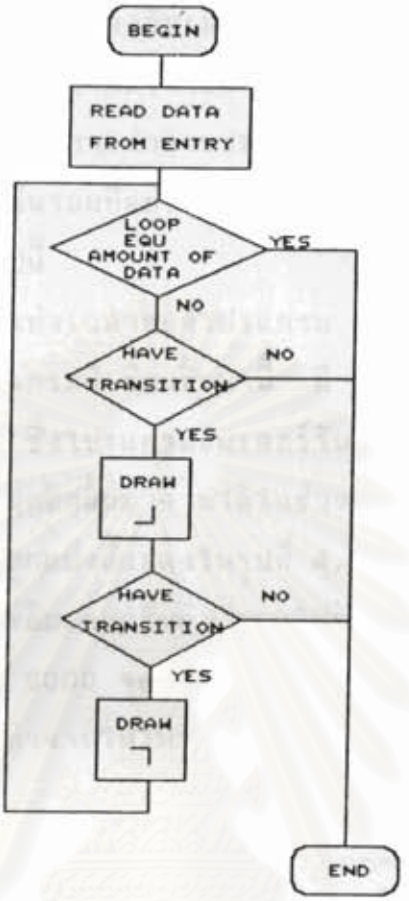


รูปที่ 4.19 โครงสร้างโปรแกรมในเครื่องไมโครคอมพิวเตอร์

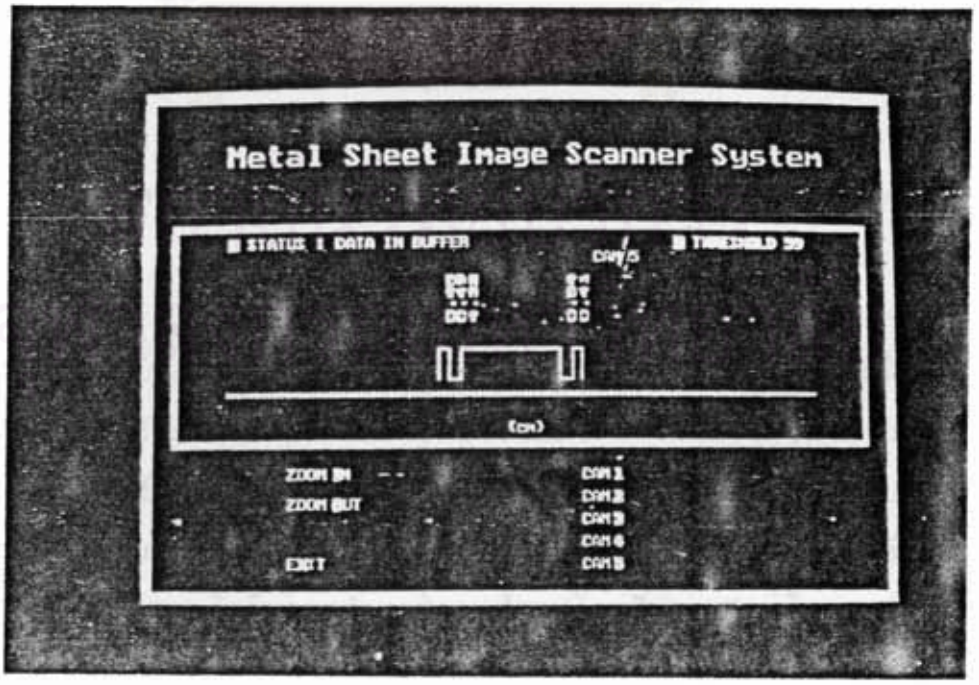
รูปที่ 4.20 โครงสร้างของข้อมูลที่ได้จากการสแกน

4.2.2.2 โปรแกรมการแสดงผลกราฟฟิก(Monitor Program)

หลักจากส่งคำสั่ง MONITORING แล้ว เครื่องไมโครคอมพิวเตอร์ก็จะทำหน้าที่รับข้อมูลจากเครื่องอิมเมจสแกนเนอร์มาทำการแสดงผลเป็นกราฟฟิก เพื่อช่วยแสดงผลการปรับแต่งให้กับผู้ใช้เมื่อต้องการติดตั้งระบบใหม่ การแสดงผลจะทำโดยการวาดภาพตัดขวางของแผ่นโลหะ และขนาดต่างๆ ด้วย โดยโปรแกรมนั้นแสดงในรูปที่ 4.21 และ รูปตัวอย่างการแสดงผลแสดงในรูปที่ 4.22



รูปที่ 4.21 โปรแกรมการแสดงผลกราฟฟิก(Monitor Program)



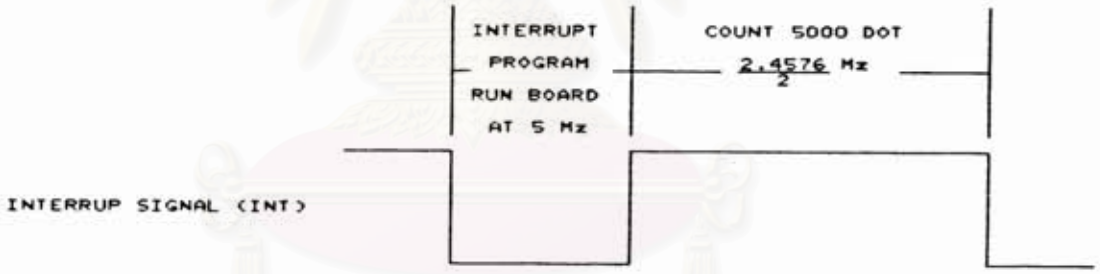
รูปที่ 4.22 รูปแบบการแสดงผลกราฟฟิก

4.2.3 โปรแกรมในส่วนของเครื่องอิมเมจสแกนเนอร์

เครื่องอิเมจสแกนเนอร์ที่สร้างขึ้นนี้ ทำงานโดยรับคำสั่งจากเครื่องไมโครคอมพิวเตอร์เท่านั้น แต่การทำงานของส่วนฮาร์ดแวร์ต่างๆนั้น ถูกควบคุมโดยผ่านโปรแกรมภายในอีพรอมที่อยู่ในบอร์ดไมโครคอนโทรลเลอร์ ซึ่งจะได้อธิบายถึงส่วนต่างๆ ดังต่อไปนี้

4.2.3.1 การแบ่งเวลาของโปรแกรม

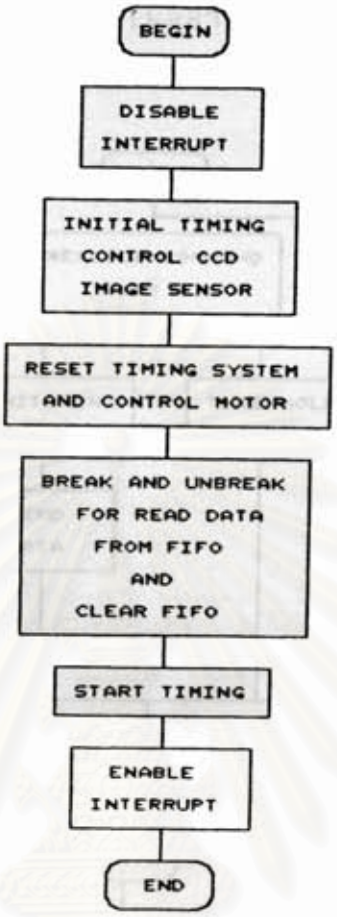
โปรแกรมที่เขียนขึ้นมานี้ มี 2 ส่วน คือ โปรแกรมหลักและโปรแกรมอินเตอร์รัพต์ ซึ่งโปรแกรมอินเตอร์รัพต์นั้นจะทำงานเป็นคาบเวลาที่แน่นอน ทำให้โปรแกรมหลักนั้นทำงานได้ในช่วงเวลาที่ว่างจากการอินเตอร์รัพต์เท่านั้น คาบเวลาที่ถูกแบ่งนี้แสดงในรูปที่ 4.23 ซึ่งในห้วงแรกนั้นจะเป็นการทำงานของโปรแกรมอินเตอร์รัพต์ และในห้วงที่สองมีระยะเวลาเท่ากับเวลาที่ใช้ในการเลื่อนข้อมูล 5000 จุด จากชุดรีจิสเตอร์ ภายในตัวอิมเมจเซ็นเซอร์ หลังจากนั้นก็กลับมาทำงานในโปรแกรมอินเตอร์รัพต์



รูปที่ 4.23 การแบ่งเวลาของโปรแกรมในเครื่องอ่านภาพแผ่นโลหะเรียบ

4.2.3.2 โปรแกรมอินเตอร์รัพต์

การทำงานของโปรแกรมอินเตอร์รัพต์นี้ เริ่มต้นด้วยการดิสเอเบิลอินเตอร์รัพต์เพื่อไม่ให้เกิดอินเตอร์รัพต์ซ้ำ จากนั้นจึงตั้งค่าดาวน์เคาน์เตอร์ เพื่อสร้างสัญญาณไทม์มิ่งสำหรับควบคุมตัวอิมเมจเซ็นเซอร์ และโปรแกรมอินเตอร์รัพต์ยังควบคุมการทำงานของสเต็บปิ้งมอเตอร์ การอ่านข้อมูลเข้าหน่วยความจำ FIFO และเริ่มต้นการทำงานของส่วนสร้างสัญญาณไทม์มิ่ง ก่อนออกจากโปรแกรมอินเตอร์รัพต์จึงอีเนเบิลอินเตอร์รัพต์ เพื่อยอมให้มีการอินเตอร์รัพต์ต่อไป โปรแกรมดังแสดงในรูปที่ 4.24



รูปที่ 4.24 โปรแกรมอินเตอร์รัพต์

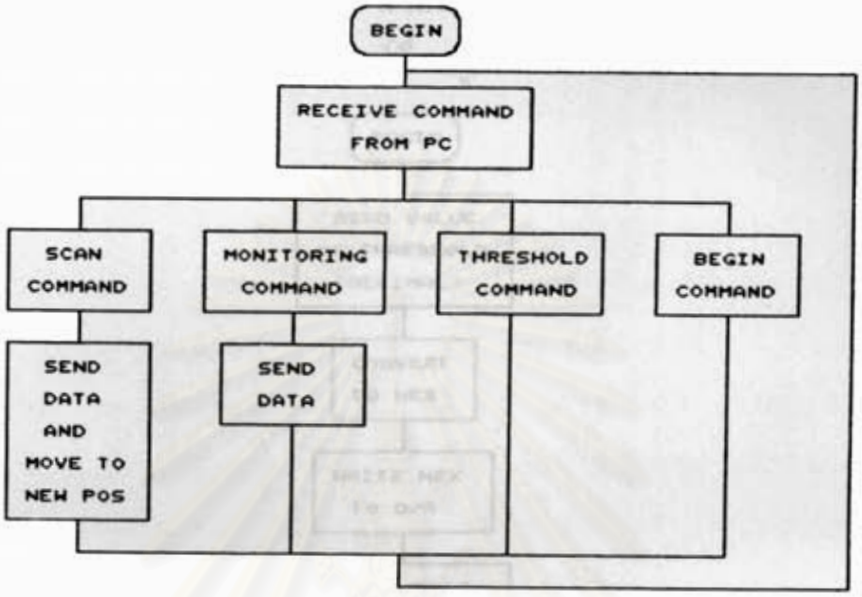
4.2.3.3 โครงสร้างโปรแกรมหลักในเครื่องอ่านภาพแผ่นโลหะเรียบ

จากรูปที่ 4.25 ในส่วนแรก ๆ คือ การรับค่าสั่งที่ส่งมาจากเครื่องไมโครคอมพิวเตอร์ ซึ่งก็ได้กล่าวมาแล้ว และ หลังจากนั้นจะทำงานตามค่าสั่งที่ได้เลือก โดยมี 5 ค่าสั่ง คือ SCAN,MONITORING,THRESHOLD, BEGIN ค่าส่ง MONITORING นั้นจะเป็นเพียงการส่งข้อมูลเพียงอย่างเดียว โดยโครงสร้างของโปรแกรมทั้งหมดดังแสดงในรูปที่ 4.25 ส่วนค่าสั่ง SCAN, THRESHOLD และ BEGIN นั้นมีรายละเอียดดังจะได้กล่าวต่อไปหลังจากนี้

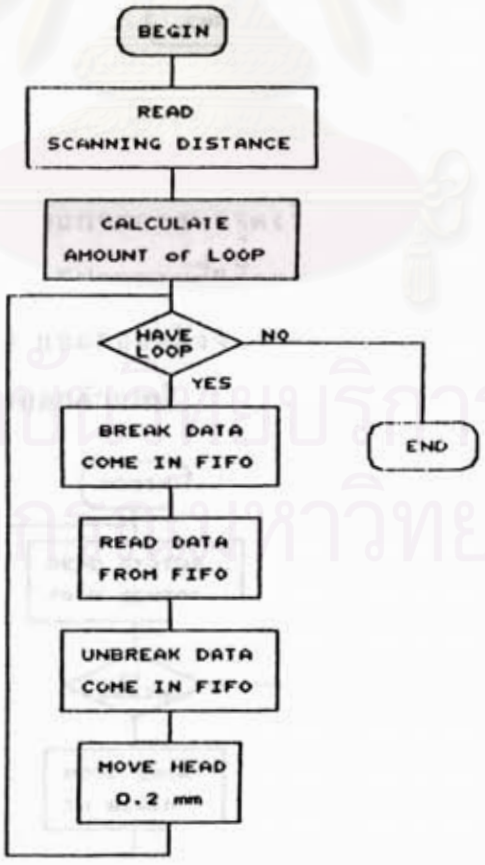
4.2.3.4 โปรแกรมการสแกน

เริ่มต้นการทำงานของโปรแกรม โดยรับค่าระยะทางการสแกนจากผู้ใช้ เพื่อนำมาคำนวนหาจำนวนเส้นของการสแกน หลังจากนั้นก็หยุดการเก็บข้อมูลเข้าหน่วยความจำ FIFO และ ทำการถ่ายเทข้อมูลไปยังแรมในบอร์ดไมโครคอนโทรลเลอร์ เมื่อถ่ายเทเสร็จจึงเริ่มการเก็บข้อมูลต่อไป และ

เลื่อนหัวอ่านไปอีก 0.2 มิลลิเมตร เพื่ออ่านข้อมูล ณ ตำแหน่งใหม่ ทำซ้ำจนกระทั่งครบจำนวนเส้นสแกน โปรแกรมดังแสดงในรูปที่ 4.26



รูปที่ 4.25 โครงสร้างโปรแกรมหลักในเครื่องอ่านภาพแผ่นโลหะเรียบ



รูปที่ 4.26 โปรแกรมการสแกน

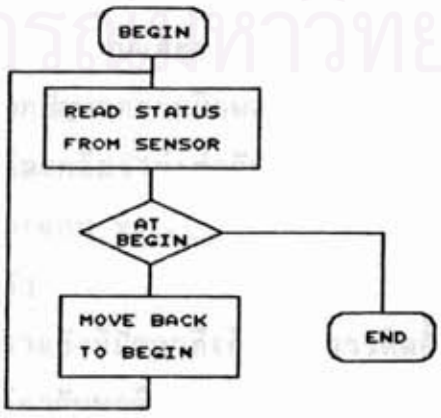
4.2.3.5 โปรแกรมเปลี่ยนค่าเทรสโฮลด์

ทำหน้าที่อ่านค่าเทรสโฮลด์ที่เป็นเลขฐาน10 จากคีย์บอร์ดเข้ามาเปลี่ยนเป็นเลขฐาน 16 จากนั้นจึงเขียนข้อมูลที่เป็นเลขฐาน 16 นี้ไปยัง D/A converter ค่าเทรสโฮลด์ก็จะถูกเปลี่ยนเป็นค่าใหม่ โปรแกรมแสดงในรูปที่ 4.27

รูปที่ 4.27 โปรแกรมเปลี่ยนค่าเทรสโฮลด์

4.2.3.6 โปรแกรมการหาจุดเริ่มต้นของการสแกน

เป็นโปรแกรมที่ใช้ตรวจสอบแบบง่ายๆ โดยใช้ลิมิตสวิตช์ในการตรวจสอบตำแหน่ง และถอยหลังจนกระทั่งถึงตำแหน่งเริ่มต้นจึงหยุด ซึ่งที่จุดเริ่มต้นจะต้องติดตั้งตัวบอกตำแหน่ง โปรแกรมดังแสดงในรูปที่ 4.28



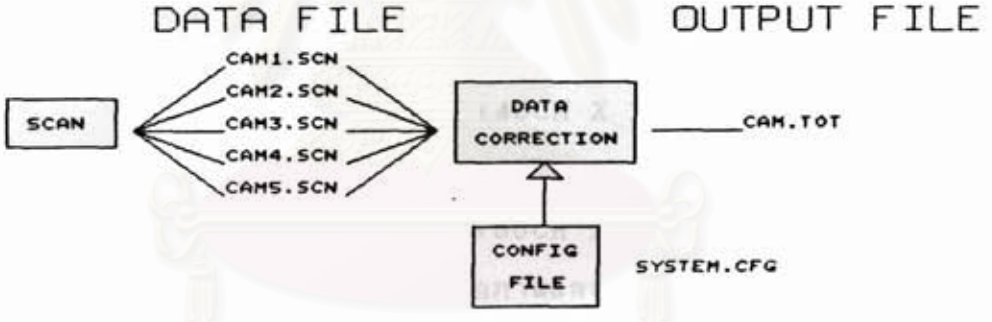
รูปที่ 4.28 โปรแกรมการหาจุดเริ่มต้นของการสแกน

4.2.4 โปรแกรมรวมภาพและแก้ไขภาพ

เนื่องจากขนาดความกว้างของภาพที่ค่อนข้างมาก ประกอบกับความแม่นยำที่ต้องการ และเหตุผลอีกหลายๆอย่างดังที่กล่าวในบทที่ 3 ทำให้ต้องใช้กล้องถึง 5 กล้อง ซึ่งก็ทำให้การต่อภาพจากกล้องทุกตัวทำได้ยาก แต่อย่างไรก็ตามยังมีวิธีการที่สามารถทำได้ โดยจะอธิบายดังต่อไปนี้

4.2.4.1 ระบบไฟล์ที่ใช้

จากการที่ข้อมูลที่ได้ยังไม่สามารถรวมกันได้ในทันที เนื่องจากข้อมูลจะต้องทำการแก้ไขก่อนที่จะนำมารวมกัน เพื่อความสะดวกและได้ประสิทธิภาพ จะทำการแก้ไขต่างๆที่เครื่องไมโครคอมพิวเตอร์ ดังนั้นการเก็บภาพจากการสแกนควรแยกออกเป็น 5 ไฟล์ คือ CAM1.SCN..CAM5.SCN ดังแสดงในรูปที่ 4.29 โดยค่าคงที่ต่างๆที่นำมาใช้ในโปรแกรมรวมภาพ และแก้ไขภาพนั้นจะอยู่ในไฟล์ชื่อ SYSTEM.CFG หลังจากการแก้ไข และรวมภาพแล้ว ก็จะเก็บภาพไว้ในเอาต์พุตไฟล์ของระบบ ชื่อ CAM.TOT



รูปที่ 4.29 ระบบไฟล์ที่ใช้

4.2.4.2 การต่อภาพและการแก้ไขความผิดเพี้ยนจากการสแกน

จากปัญหาความผิดพลาดที่เกิดขึ้นในระบบการสแกนภาพเช่น ขนาดของภาพที่ได้จากแต่ละกล้องไม่เท่ากัน เนื่องจากอัตราขยายไม่เท่ากัน ซึ่งมีผลในแนวของไลน์สแกน(แกน x) ดังนั้นจึงต้องแก้ไขโดยการคูณค่า X_c แฟกเตอร์สำหรับกล้องแต่ละตัว

แต่อย่างไรก็ตามยังมีปัญหาที่เกิดจากการติดตั้ง ซึ่งไม่สามารถทำให้ตัวเซ็นเซอร์อยู่ในแนวเดียวกันพอดีได้(แกน y) ก็จะใช้ค่า Y_{shift} เพื่อเลื่อนให้อยู่ในระดับเดียวกัน

ส่วนการต่อภาพระหว่างตัวอิมเมจเซ็นเซอร์ก็เช่นกัน ที่ไม่สามารถวางต่อกันได้พอดี โดยจะต้องวางตัวเซ็นเซอร์เหลื่อมกันตรงรอยต่อเล็กน้อยเพื่อใช้สำหรับการต่อภาพ ซึ่งจะกำหนดไว้ 2 ที่ คือ บริเวณข้างหน้าของข้อมูลในแถว(เป็นตำแหน่งO_{front}) และ บริเวณข้างหลังของข้อมูลในแถว(เป็นตำแหน่ง O_{back}) โดยใช้ค่า O_{front} และ O_{back} มาเปรียบเทียบ ถ้าข้อมูล X ใดมีค่ามากกว่า O_{back} ก็จะตัดทิ้ง และ สำหรับค่า O_{front} จะนำไปลบออกจากข้อมูล X ทุกตัว และถ้าข้อมูลตัวใดมีค่าเป็นลบก็ให้ตัดทิ้ง ดังแสดงในรูปที่ 4.30

ค่า O_{back} นั้นได้มาจากการคำนวน ซึ่งมีค่าเท่ากับ O_{front} บวกกับ (20CM X DOT PER CM) ส่วนค่า O_{front} จะกำหนดขึ้นเอง ในด้านการใช้งานนั้นเราจะเก็บค่าตัวแปรเหล่านี้ไว้ในไฟล์คอนฟิกของระบบ คือ SYSTEM.CFG ซึ่งจะถูกอ่านขึ้นมาจากไฟล์ในตอนที่ทำกระบวนการแก้ไขข้อมูล

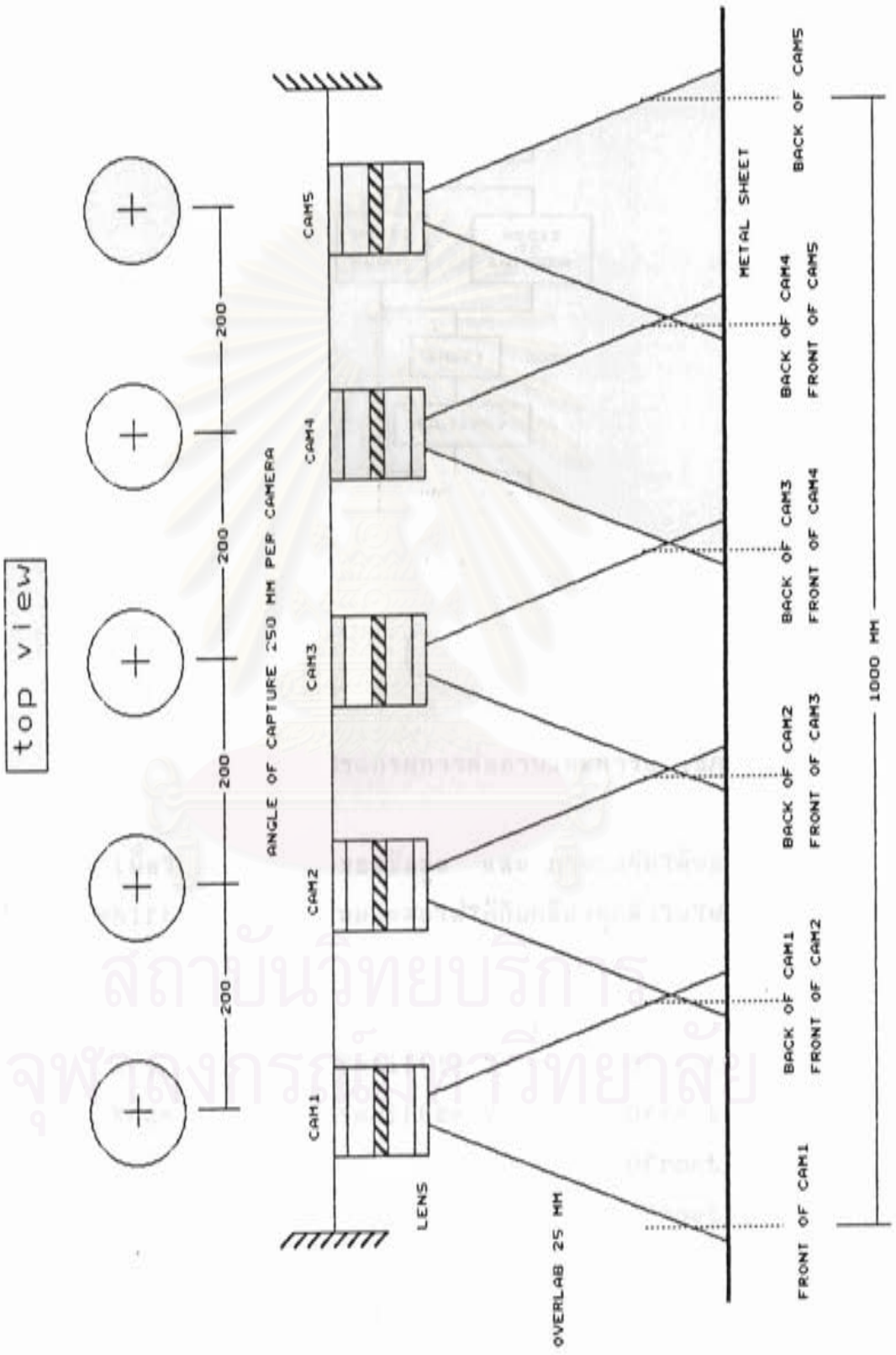
ตำแหน่งที่ใช้อ้างอิงในการต่อภาพ โดยเป็นค่าที่ใช้ในการบวกให้กับกล้องแต่ละตัว มีค่าดังต่อไปนี้

ข้อมูลกลอ้งตัวที่ 1 : + (0CM X DOT PER CM)
ข้อมูลกลอ้งตัวที่ 2 : + (20CM X DOT PER CM)
ข้อมูลกลอ้งตัวที่ 3 : + (40CM X DOT PER CM)
ข้อมูลกลอ้งตัวที่ 4 : + (60CM X DOT PER CM)
ข้อมูลกลอ้งตัวที่ 5 : + (80CM X DOT PER CM)

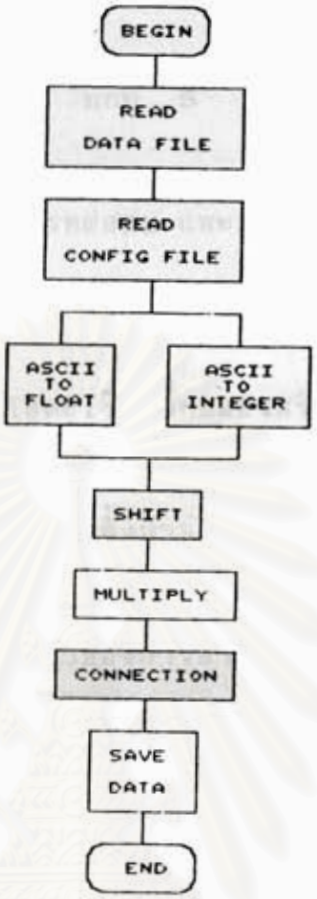
4.2.4.3 โปรแกรมการต่อภาพและการแก้ไขภาพ

จากหลักการในการต่อภาพดังที่กล่าวมาแล้ว จึงได้นำมาเขียนโปรแกรม โดยนำค่าต่างๆที่อยู่ในไฟล์ SYSTEM.CFG คือ Xc Yshift และ Ofront มาใช้ในโปรแกรม เพื่อปรับแต่งให้ข้อมูลที่ได้มีความถูกต้องแม่นยำ

การทำงานของโปรแกรม เริ่มด้วยการอ่านภาพที่ได้จากการสแกน จากนั้นก็อ่านไฟล์ SYSTEM.CFG แล้วทำการเปลี่ยนจากตัวเลขในรหัสอัสกี้ให้เป็นเลขมีจุดทศนิยม(FLOAT) และเลขจำนวนเต็ม(INTEGER) เมื่อได้ค่าต่างๆแล้ว จึงเริ่มกระบวนการดังนี้ตามลำดับเลื่อนเส้นสแกน(SCAN LINE SHIFT) แล้วคูณค่าแฟกเตอร์(FACTOR MULTIPLY)เพื่อปรับเปลี่ยนขนาดของภาพ และจากนั้นก็เข้าสู่กรรมวิธีการต่อภาพ(CONNECTION) หลักจากกระบวนการทุกอย่างเสร็จสิ้นแล้วจึงทำการเก็บข้อมูลที่แก้ไขแล้วนี้ในเอาต์พุตไฟล์ CAM.TOT โปรแกรมการต่อภาพและการแก้ไขภาพ ดังแสดงในรูปที่ 4.31



รูปที่ 4.30 การต่อภาพที่ได้จากกล้อง 5 ตัว



รูปที่ 4.31 โปรแกรมการต่อภาพและการแก้ไขภาพ

เพื่อให้ความแม่นยำของข้อมูล และ ภาพต่อกันได้พอดี จะต้องนำค่า ค่า Xc Yshift Ofront ที่เหมาะสมใส่ให้กับกล้องทุกตัวในไฟล์ SYSTEM.CFG ซึ่งค่าที่จะต้องใส่มีดังต่อไปนี้

Xc1= ?	Yshift1= ?	Ofront1= ?
Xc2= ?	Yshift2= ?	Ofront2= ?
Xc3= ?	Yshift3= ?	Ofront3= ?
Xc4= ?	Yshift4= ?	Ofront4= ?
Xc5= ?	Yshift5= ?	Ofront5= ?

นอกจากนี้ยังได้อธิบายถึงวิธีการปรับเทียบ และทำการทดลองต่างๆอีก ดังจะได้กล่าวต่อไปในบทที่ 5

บทที่ 5

การปรับเทียบ การทดสอบ และการวิเคราะห์ผล

5.1 การปรับเทียบ

เครื่องอ่านภาพแผ่นโลหะเรียบที่ทดลองสร้างขึ้นนี้ ต้องการความแม่นยำในการประกอบชิ้นส่วนทางกลต่างๆทั้งหมด จึงจะสามารถอ่านภาพได้ถูกต้องตามสเกล แต่การประกอบและติดตั้งหัวอ่านที่มี กล้อง และ CCD ไม่สามารถติดตั้งได้อย่างแม่นยำ จึงจำเป็นจะต้องปรับแต่งระบบให้มีความถูกต้องก่อน การคลาดเคลื่อนที่จะมีผลต่อความแม่นยำของการอ่านภาพ ได้แก่ อัตราขยายของกล้อง 5 กล้อง มีอัตราแตกต่างกัน ตำแหน่งการติดตั้งกล้องไม่อยู่ในแนวเส้นตรงเดียวกัน และอาจจะเอียงจากเส้นสแกน นอกจากนั้นขอบเขตการแบ่งอาณาเขตระหว่างกล้องแต่ละตัวซึ่งจะรับภาพเหลื่อมกันก็จะมีความสำคัญด้วย เนื่องจากจะทำให้ภาพไม่ต่อกันสนิทเมื่อนำภาพจาก 5 กล้อง มารวมเป็นภาพเดียวกัน

การปรับเทียบเครื่องอ่านภาพแผ่นโลหะนี้ จะใช้กรรมวิธีทางด้านซอฟต์แวร์เข้าช่วย โดยไม่ต้องไปปรับตำแหน่งของกล้องแต่อย่างใด จะอ่านข้อมูลดิบจากกล้องทุกตัวเข้ามาในคอมพิวเตอร์ จากนั้นนำข้อมูลเหล่านั้นมาปรับใหม่ ซึ่งจะต้องคำนวณโดยใช้ข้อมูลการปรับแต่งของกล้องแต่ละตัว ข้อมูลการปรับแต่งนี้จะเก็บไว้ในไฟล์ชื่อ SYSTEM.CFG และ ตัวเลขจะได้จากการทดลองการปรับแต่ง ซึ่งมีกรรมวิธีดังต่อไปนี้

1) เตรียมแบบมาตรฐานสำหรับการสแกนอ่าน เพื่อใช้ในการปรับเทียบ คือ แบบพิมพ์รูปวงกลมเส้นผ่าศูนย์กลาง 100 มม. $\pm$ 0 ตามที่แสดงในรูปที่ 5.3 และ แบบแผ่นโลหะเจาะวงกลม 5 วง ตามในรูปที่ 5.5

2) สแกนภาพแบบพิมพ์รูปวงกลมโดยใช้กล้อง 5 กล้อง และ จัดเก็บข้อมูลไว้ในฮาร์ดดิสก์ การสแกนอ่านภาพใช้โปรแกรม ชื่อ IBMSCAN.EXE

3) สแกนภาพแบบแผ่นโลหะเจาะวงกลม 5 วง ตามรูปที่ 5.5 โดยต้องจัดวางตำแหน่งของแผ่นโลหะให้วงกลมทั้ง 5 อยู่ชนานกับเส้นสแกน สแกนอ่านภาพด้วยกล้องทั้ง 5 จากนั้นรวมภาพจากที่ได้จากกล้อง 5 ชุด โดยโปร

แกรมรวมและแก้ไขภาพชื่อ ADDFILE.EXE

4) นำภาพที่รวมแล้วเข้าไปในโปรแกรม Autocad

5) วัดขนาดของภาพที่ได้จากการสแกนโดยโปรแกรม Autocad

6) เปลี่ยนค่า Xc Yshift และ Ofront ในไฟล์ SYSTEM.CFG โดยเริ่มที่การปรับอัตราขยายภาพ ด้วยการเปลี่ยนค่า Xc ซึ่งถ้าขนาดที่ได้มากกว่าค่าที่ถูกต้องก็ให้ลดค่า Xc และ ถ้าขนาดที่ได้น้อยกว่าค่าที่ถูกต้องก็ให้เพิ่มค่า Xc แต่ในช่วงที่ทำการปรับ Xc จะคงค่า Ofront และ Yshift เอาไว้ที่ค่าโดยปริยาย(default)คือ 500 และ 0 ตามลำดับจนกว่าจะได้ค่า Xc ที่ถูกต้อง ซึ่งการปรับค่า Xc จะใช้แบบพิมพ์วงกลม ดังรูปที่ 5.3

7) ปรับค่า Yshift ซึ่งเป็นค่าปรับระดับให้กล้องวางอยู่ในแนวสแกนเดียวกัน การปรับใช้ภาพแผ่นโลหะวงกลม 5 วง ในรูปที่ 5.6 โดยให้รูปวงกลมมาอยู่ในแนวเดียวกัน ซึ่งถ้าค่า Yshift มากเกินไปรูปวงกลมจะเลยไปไม่อยู่ในแนวเดียวกัน แต่ถ้าค่ามากเกินไปรูปวงกลมก็จะอยู่หลังรูปวงกลมอันอื่นๆ ปรับค่า Yshift ของกล้องแต่ละตัวจนกระทั่งได้ค่า Yshift ที่ถูกต้อง

8) การเปลี่ยนค่า Ofront ของกล้องแต่ละตัว ซึ่งเป็นค่าที่ใช้ในการแบ่งอาณาเขตของกล้องแต่ละตัว ค่า Ofront คือ ระยะจากจุดเริ่มต้นการสแกนถึงจุดที่เริ่มนับข้อมูลของกล้องนั้น (ดูรูปที่ 4.30 ประกอบ) การปรับค่า Ofront นี้ จะต้องสแกนภาพแผ่นโลหะเจาะวงกลม 5 วง โดยตั้งให้เอียง และภาพแผ่นโลหะเจาะช่องสี่เหลี่ยมดังตัวอย่างในรูปที่ 5.7 และ 5.8 วิธีปรับค่า Ofront ของแต่ละกล้องจนรอยต่อระหว่างกล้องต่อกันสนิท โดยสังเกตที่ขอบของแผ่นโลหะ ขอบของสี่เหลี่ยมจะต้องต่อเชื่อมกันสนิท ถ้าค่า Ofront มากหรือ น้อยเกินไปภาพที่ถูกตัดมาจะไม่ต่อเนื่องกันทำให้รอยต่อของภาพไม่สนิท

9) กระทำซ้ำวิธีเดิมตั้งแต่ 1 ถึง 6 จนกระทั่งได้ค่าที่ถูกต้องแม่นยำ และ ภาพที่อ่านได้จากกล้อง 5 ชุดต่อกันสนิท

จากการทดลองการปรับค่าทั้ง 3 ดังที่กล่าวมาแล้ว เพื่อให้รูปภาพต่อกันได้สนิท และ ระยะต่างๆในภาพมีความแม่นยำที่สุด ทำให้ได้ค่าต่างๆในไฟล์ SYSTEM.CFG ดังต่อไปนี้

Xc1=1.0315	Yshift1=8	Ofront1=491
Xc2=1.0315	Yshift2=2	Ofront2=491
Xc3=1.0355	Yshift3=0	Ofront3=509

Xc4=1.0380 Yshift4=3 Ofront4=480

Xc5=1.0385 Yshift5=3 Ofront5=494

5.2 การทดสอบการทำงานของเครื่องอ่านภาพแผ่นโลหะเรียบ

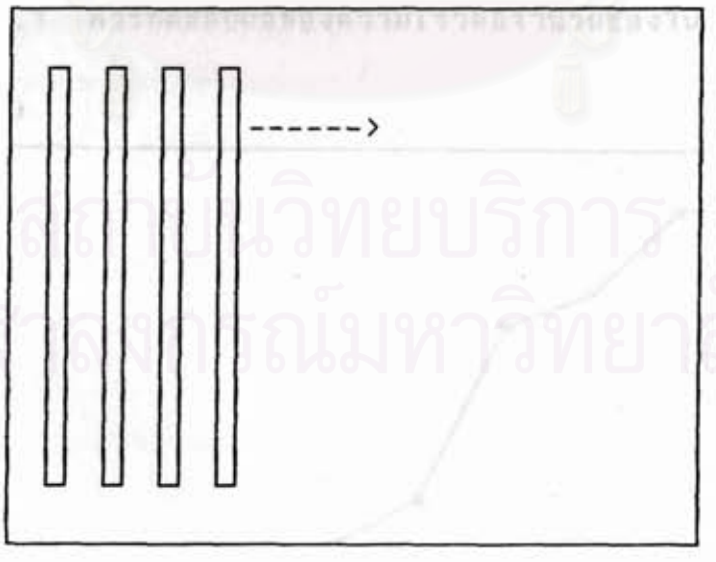
ในบทที่ 4 ได้กล่าวถึงการออกแบบ และประกอบสร้างวงจรส่วนต่างๆ ทั้งหมด ซึ่งหลังจากได้ประกอบสร้างแล้วก็นำมาทดสอบการทำงาน ดังต่อไปนี้

5.2.1 ความเร็วของการเคลื่อนที่ของหัวอ่าน

การทดสอบนี้เป็นการวัดความเร็วของส่วนกลไกเคลื่อนไหว ซึ่งทำได้โดยการสแกนแบบไม่มีแผ่นโลหะเป้าหมาย โดยมีระยะการสแกนเท่ากับ 2.4 เมตร ผลที่ได้ออกมาดังนี้

การสแกนไป 10:48 นาที การถอยกลับ 9:58 นาที

การทดสอบประสิทธิภาพของเครื่องอ่านภาพแผ่นโลหะเรียบ และ โปรแกรมการต่อภาพ ทั้งในด้านความเร็ว และ ปริมาณข้อมูลที่ได้ ซึ่งได้ใช้แผ่นภาพที่มีช่องขนาด กว้าง 10 มิลลิเมตร ยาว 150 มิลลิเมตร แล้ว ทำการสแกนและบันทึกผล จากนั้นก็เพิ่มจำนวนช่องเพื่อสังเกตุการเปลี่ยนแปลง โดยกำหนดให้ระยะการสแกนคงที่เท่ากับ 300 มิลลิเมตร ภาพที่นำมาสแกนมีลักษณะดังในรูปที่ 5.1 โดยนำไปใช้ในการทดสอบในหัวข้อ 5.2.2,5.3.1,5.3.2

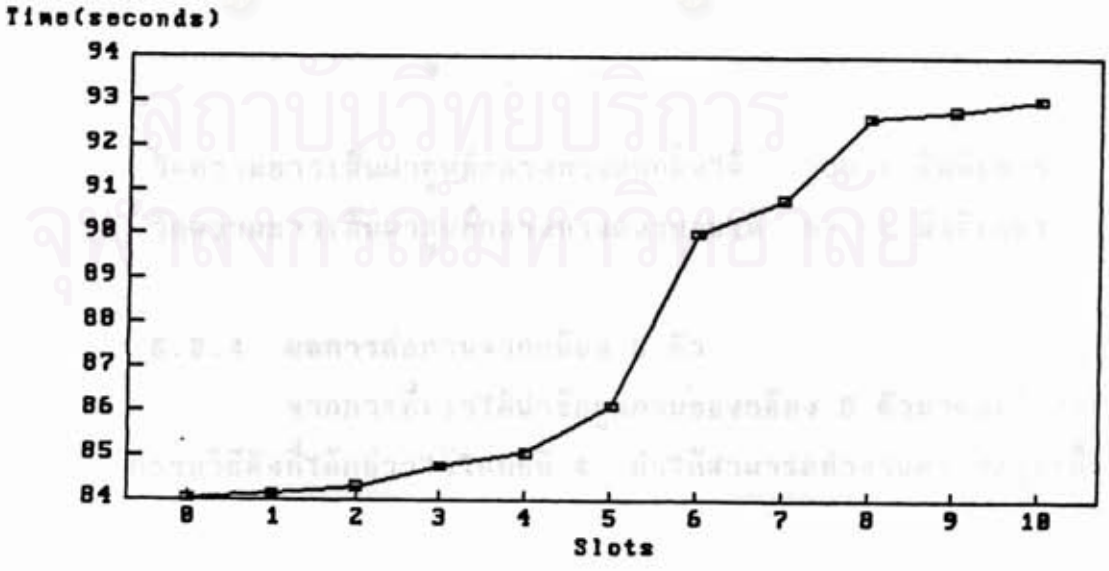


รูปที่ 5.1 ลักษณะภาพที่นำมาใช้ทำการทดลอง

5.2.2 ผลของความเร็วต่อจำนวนช่องในการสแกน (ดังแสดงในตารางที่ 5.1 และ รูปที่ 5.2)

จำนวนช่อง	เวลาที่ใช้ในการสแกน(นาที)
0	1:24.07
1	1:24.14
2	1:24.30
3	1:24.76
4	1:25.06
5	1:26.12
6	1:30.01
7	1:30.79
8	1:32.65
9	1:32.80
10	1:33.04

ตารางที่ 5.1 การทดสอบผลของความเร็วต่อจำนวนช่องในการสแกน



รูปที่ 5.2 กราฟความเร็วของการสแกน

5.2.3 ขนาดความถูกต้องที่ได้จากการสแกน

หลังจากการปรับเทียบในหัวข้อ 5.1 จึงเริ่มทดสอบโดยการสร้างรูปวงกลม ที่กำหนดขนาดไว้แน่นอน แล้วนำไปสแกน เมื่อได้ข้อมูลเป็นที่เรียบร้อยแล้ว จึงทำการวัดขนาดในโปรแกรม Autocad และ บันทึกผล รวมทั้งแสดงรูปภาพที่ได้จากการสแกนเพื่อใช้ในการเปรียบเทียบกัน ตัวอย่างค่าที่วัดได้ ดังแสดงในรูปที่ 5.3 และ รูปที่ 5.4

ต้นแบบ มีเส้นผ่าศูนย์กลาง 100 มิลลิเมตร

กล้องที่ 1

วัดความยาวเส้นผ่าศูนย์กลางทางแนวตั้งได้ 100.2 มิลลิเมตร

วัดความยาวเส้นผ่าศูนย์กลางทางแนวนอนได้ 99.70 มิลลิเมตร

กล้องที่ 2

วัดความยาวเส้นผ่าศูนย์กลางทางแนวตั้งได้ 100.4 มิลลิเมตร

วัดความยาวเส้นผ่าศูนย์กลางทางแนวนอนได้ 99.85 มิลลิเมตร

กล้องที่ 3

วัดความยาวเส้นผ่าศูนย์กลางทางแนวตั้งได้ 100.2 มิลลิเมตร

วัดความยาวเส้นผ่าศูนย์กลางทางแนวนอนได้ 99.9 มิลลิเมตร

กล้องที่ 4

วัดความยาวเส้นผ่าศูนย์กลางทางแนวตั้งได้ 100.2 มิลลิเมตร

วัดความยาวเส้นผ่าศูนย์กลางทางแนวนอนได้ 99.75 มิลลิเมตร

กล้องที่ 5

วัดความยาวเส้นผ่าศูนย์กลางทางแนวตั้งได้ 100.6 มิลลิเมตร

วัดความยาวเส้นผ่าศูนย์กลางทางแนวนอนได้ 99.95 มิลลิเมตร

5.2.4 ผลการต่อภาพจากกล้อง 5 ตัว

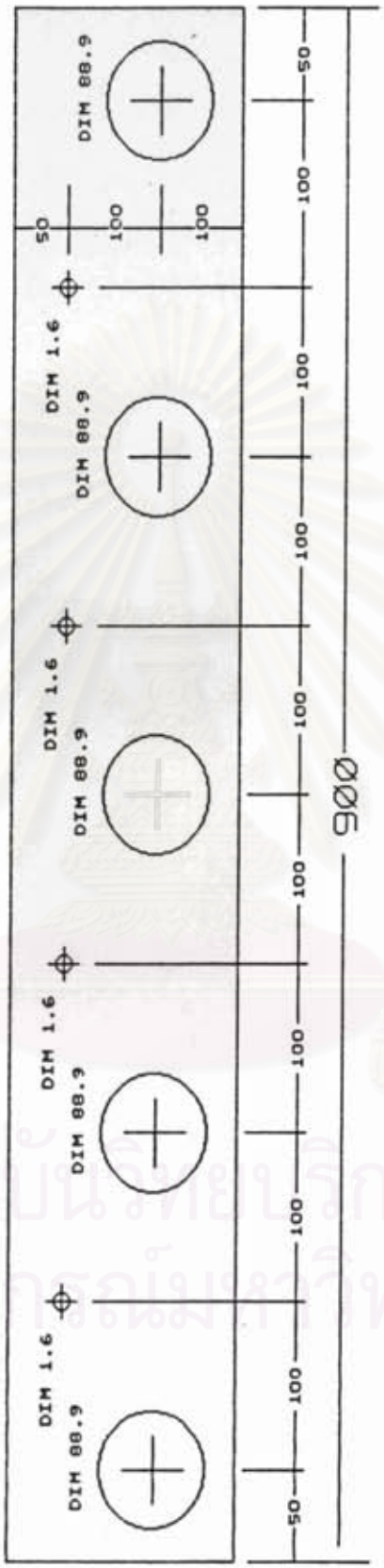
จากการที่เราได้นำข้อมูลภาพของกล้อง 5 ตัวมาต่อเข้าด้วยกัน โดยกรรมวิธีดังที่ได้กล่าวไว้ในบทที่ 4 ทำให้สามารถทำงานครอบครุมเนื้อที่ได้มากขึ้น ดังจะแสดงรายละเอียดของต้นแบบ และผลที่ได้จากการสแกน ในรูปที่ 5.5 ถึง รูปที่ 5.8

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

รูปที่ 5.3 รูปวงกลมต้นแบบ

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

รูปที่ 5.4 รูปวงกลมที่ได้จากการสแกน

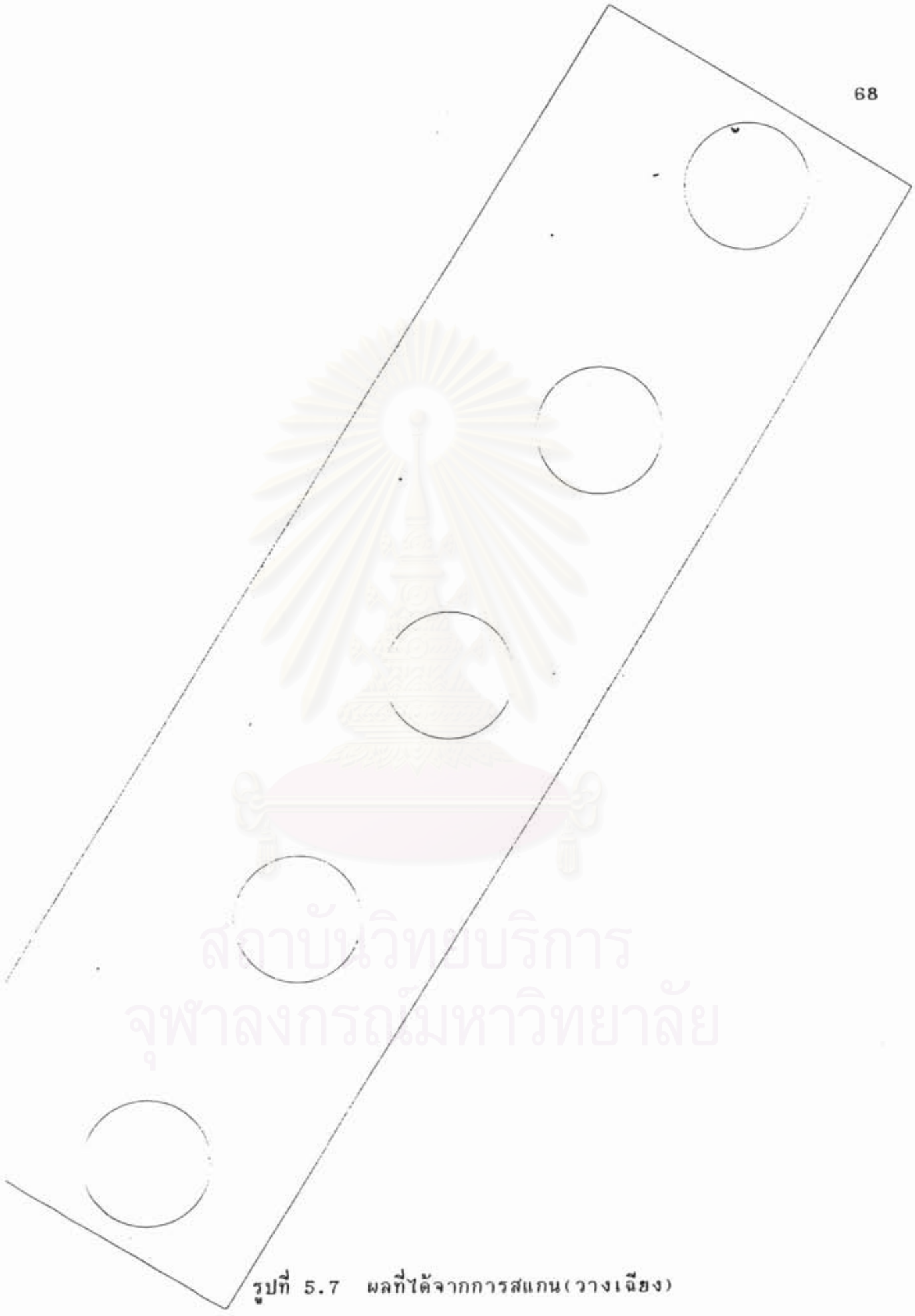


รูปที่ 5.5 แบบของแผ่นโลหะที่นำมาสแกน

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

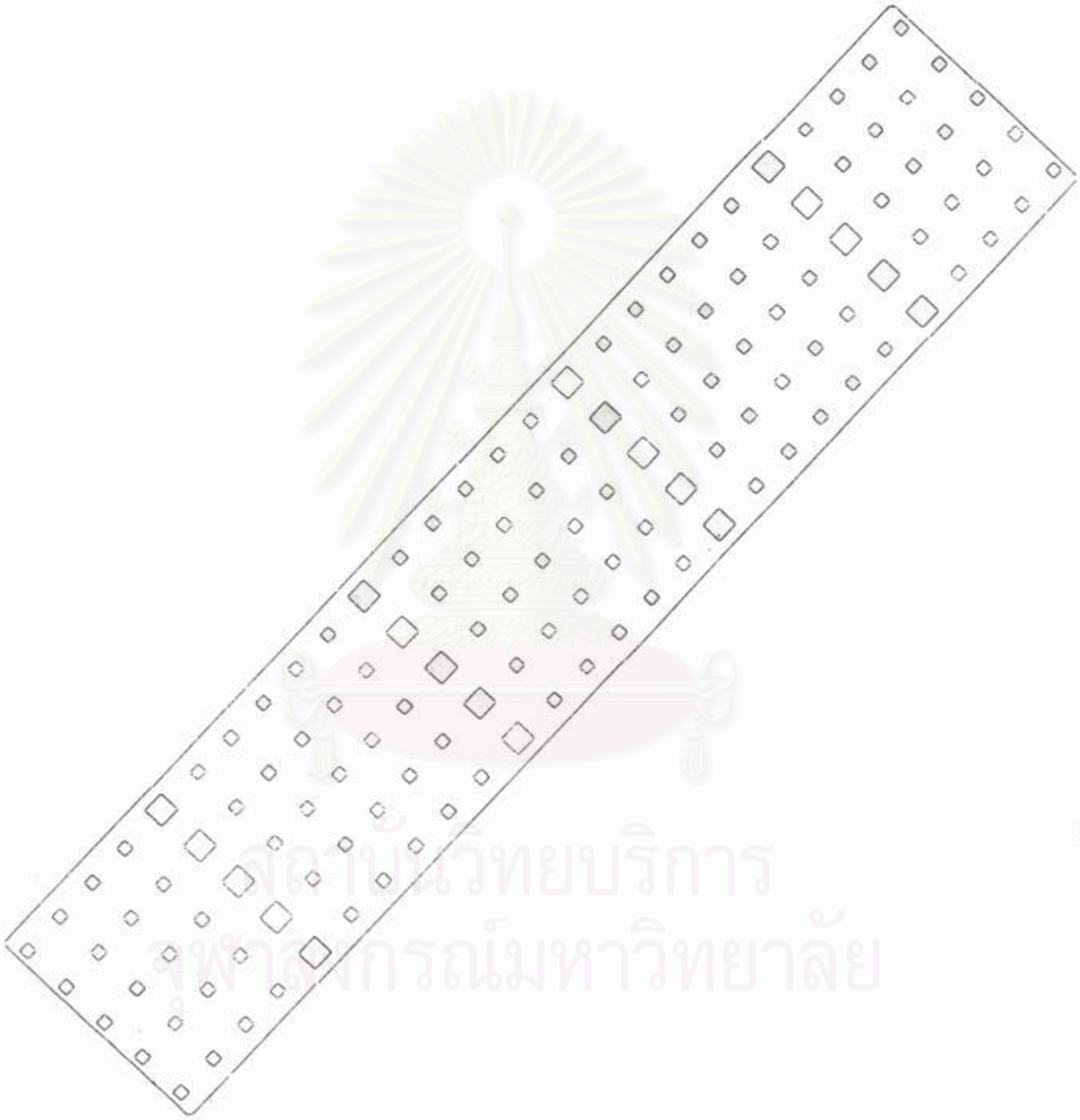
รูปที่ 5.6 ผลที่ได้จากการสแกน(วางตรง)

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 5.7 ผลที่ได้จากการสแกน(วางเฉียง)

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 5.8 ตัวอย่างผลอื่นๆที่ได้จากการสแกน

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

5.3 การทดสอบโปรแกรมแก้ไขภาพและการรวมภาพ

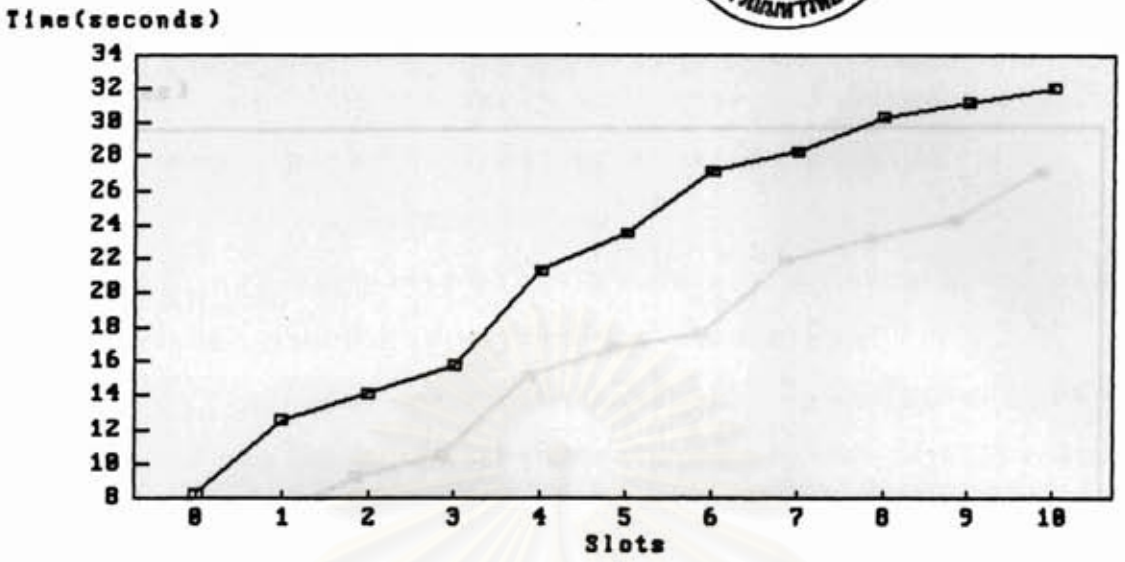
หลังจากที่ได้ทดสอบการทำงานของเครื่องอ่านภาพแผ่นโลหะเรียบแล้ว ต่อไป ก็จะกล่าวถึงในส่วนของซอฟต์แวร์ที่ใช้สำหรับแก้ไขภาพ ให้มีความ ถูกต้อง แม่นยำ และต่อภาพที่ได้จากกล้อง 5 ตัวเข้าด้วยกัน โดยจะทดสอบประสิทธิภาพ ของโปรแกรม

5.3.1 ผลของความเร็วต่อจำนวนช่องในการแก้ไขภาพและรวมภาพ (ดังแสดงในตารางที่ 5.2 และ รูปที่ 5.9)

จำนวนช่อง	เวลาที่ใช้ในการรวมข้อมูล(วินาที)
0	8.25
1	12.55
2	14.08
3	15.75
4	21.32
5	23.54
6	27.15
7	28.20
8	30.32
9	31.18
10	32.02

ตารางที่ 5.2 ผลของความเร็วต่อจำนวนช่องในการแก้ไขภาพและการรวมภาพ

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

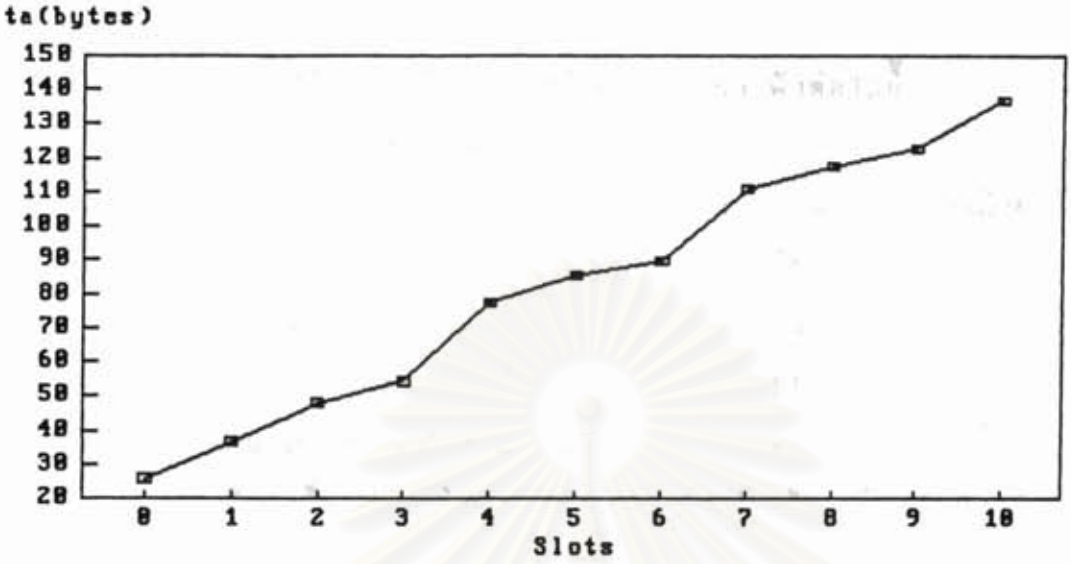


รูปที่ 5.9 ความเร็วในการแก้ไขภาพและการรวมภาพ

5.3.2 ปริมาณของข้อมูลต่อจำนวนช่อง (ดังแสดงในตารางที่ 5.3 และ รูปที่ 5.10)

จำนวนช่อง	ปริมาณของข้อมูล(Byte)
0	25983
1	36820
2	47996
3	54331
4	77414
5	85537
6	89645
7	110893
8	117365
9	122668
10	136822

ตารางที่ 5.3 ปริมาณของข้อมูลต่อจำนวนช่อง



รูปที่ 5.10 ปริมาณของข้อมูลที่ได้จากการสแกน

5.4 การวิเคราะห์ผล

1) จากผลการทดสอบความเร็วของ เครื่องอ่านภาพแผ่นโลหะเรียบ ในการการทดสอบข้อ 5.2.1 และ 5.2.2 จะเห็นได้ว่าความเร็วการสแกนขึ้นกับความซับซ้อนของภาพที่สแกน

2) จากผลการทดสอบเกี่ยวกับความถูกต้อง แม่นยำและการต่อข้อมูลภาพ ในการทดสอบข้อ 5.2.3 และ 5.2.4 พอจะสรุปผลต่างๆได้ดังต่อไปนี้ ขนาดของวงกลมจากการสแกน ผิดพลาดอยู่ในช่วง -0.3 มิลลิเมตร ถึง +0.6 มิลลิเมตร จากกล้องทั้ง 5 ตัว ซึ่งเป็นค่าที่ได้หลังจากการปรับเทียบ

ส่วนผลที่ได้จากการต่อข้อมูลภาพในรูปที่ 5.6 และ รูปที่ 5.7 ไม่สามารถสังเกตุเห็นรอยต่อ แต่ในรูปที่ 5.8 นั้นจะสามารถเห็นรอยต่อได้บ้าง ซึ่งจะได้อธิบายถึงความผิดพลาดนี้ต่อในข้อ 4

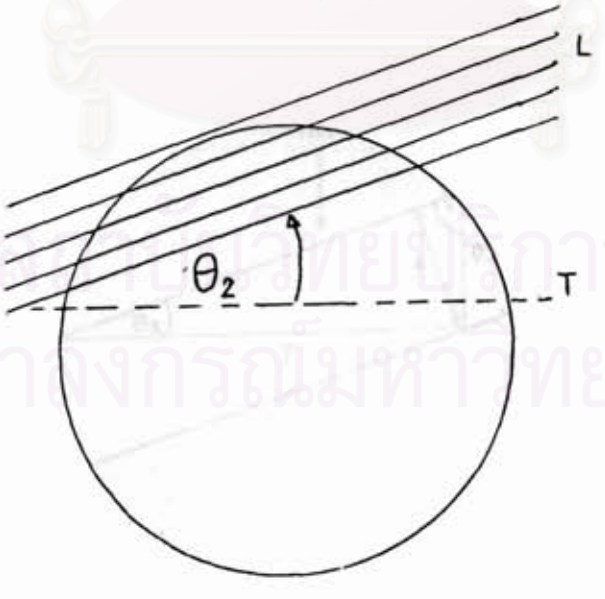
3) จากการทดสอบประสิทธิภาพของโปรแกรมแก้ไขและต่อข้อมูลภาพในข้อ 5.3.1 และ 5.3.2 ผลปรากฏว่าความเร็วการทำงานของโปรแกรมนี้ขึ้นกับปริมาณของข้อมูล ถ้าข้อมูลมีมากก็จะทำงานช้าลง อีกทั้งปริมาณของข้อมูลจะมากขึ้นตามความซับซ้อนของภาพ อย่างไรก็ตามแม้ว่าปริมาณข้อมูลจะมากขึ้นแต่ก็ไม่มากนักเมื่อเทียบกับความละเอียดของข้อมูล ซึ่งเป็นเพราะว่าการเก็บข้อมูลเฉพาะที่มีการเปลี่ยนแปลงระดับสัญญาณเท่านั้น

4) จากค่าขนาดของวงกลมที่ได้บันทึกไว้ในข้อ 5.2.3 เกิดการผิดเพี้ยนขึ้น ซึ่งก็ได้ให้การตั้งสมมุติฐานไว้ว่าการติดตั้งตัวอิมเมจเซ็นเซอร์มีความเอียงเกิดขึ้น และได้ทำการวิเคราะห์ค่าความเอียง ดังต่อไปนี้

- ความเอียงของเส้นสแกน

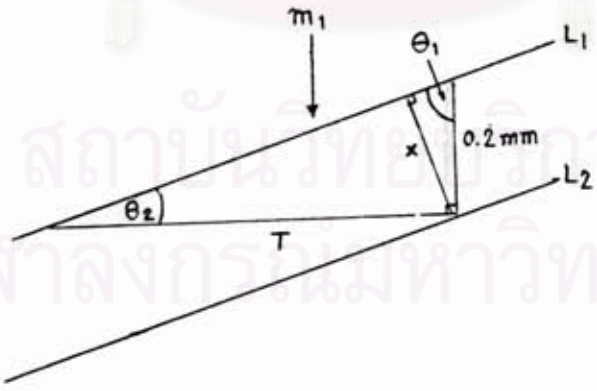
ก่อนที่จะวัดความเอียงของเส้นสแกน จะขออธิบายถึงผลที่เกิดจากการเอียง และหลักการในการคำนวนหาค่าความเอียง

เส้นสแกนที่มีความเอียงเกิดขึ้น(L) เมื่อเทียบกับเส้นระดับอ้างอิง(T) เกิดมุมเอียง θ_2 ขึ้น (ดังที่แสดงในรูปที่ 5.11) ทำให้รูปที่ได้จากการสแกนผิดเพี้ยนไปตามเส้นประในรูปที่ 5.12 จากรูปที่ 5.11 และ รูปที่ 5.12 นำมาเขียนเป็นรูปการวิเคราะห์ทางเรขาคณิต ดังรูปที่ 5.13 โดย m_1 เป็นแนวการเคลื่อนที่ของหัวอ่าน L_1 L_2 เป็นเส้นสแกนที่เอียง และเส้นอ้างอิง T โดย θ_1 คือมุมที่ทำระหว่างเส้น L_1 กับแนวการเคลื่อนที่ m_1 ซึ่งความเอียงสัมพันธ์กับเส้นผ่าศูนย์กลางของวงกลม ณ ตำแหน่งเส้นประ(ในรูปที่ 5.12) ที่ทำให้จำนวนเส้นสแกนมากขึ้น โดยมีสูตรในการหาความเอียง ดังต่อไปนี้



รูปที่ 5.11 แบบจำลองเส้นสแกนที่มีความเอียง

รูปที่ 5.12 ผลการสแกนที่ได้จากเส้นสแกนที่มีความเอียง



รูปที่ 5.13 การวิเคราะห์ทางเรขาคณิต

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

จำนวนเส้นสแกนมาตรฐาน = D_{ref} / 0.2 mm

จำนวนเส้นสแกนมากที่สุด = D_{max} / 0.2 mm

ฉะนั้น $x = D_{ref}$ / จำนวนเส้นสแกนมากที่สุด

จากรูปที่ 5.13: $x / 0.2 = \sin \theta_1$

ฉะนั้น มุม $\theta_1 = \sin^{-1} (x / 0.2)$ องศา

*** ฉะนั้น มุม $\theta_2 = 90$ องศา - θ_1 องศา

เมื่อ: x คือ ระยะห่างระหว่างเส้นสแกนในแนวเอียง (mm)

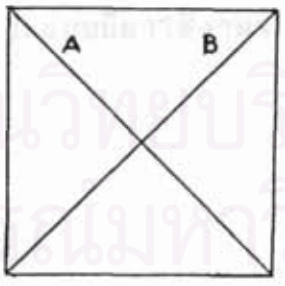
D_{ref} " ขนาดเส้นผ่าศูนย์กลางที่ถูกต้อง (mm)

D_{max} " ขนาดเส้นผ่าศูนย์ที่ผิดไป (mm)

L_{max} " จำนวนเส้นสแกนมากที่สุด (เส้น)

- การวัดทิศทางของการเอียง

การหามุมเอียงโดยวิธีแรก ยังไม่สามารถทราบได้ว่าเอียงทิศทางใด ดังนั้นจึงต้องทำขั้นตอนที่ 2 โดยการนำรูปสี่เหลี่ยมจตุรัสมาทำการสแกนดังรูปที่ 5.14 ถ้าเส้นทะแยงมุม A และ B มีค่าเท่ากันแสดงว่าเส้นสแกนไม่เอียง แต่ถ้า A ยาวกว่า B แสดงว่าเอียงทางซ้าย และ ถ้า B ยาวกว่า A แสดงว่าเอียงทางขวา



รูปที่ 5.14 รูปสี่เหลี่ยมจตุรัสนำมาทำการสแกนเพื่อหาทิศทางที่เอียง

จากวิธีดังที่ได้กล่าวมาแล้ว นำมาใช้ในการวัดความเอียงของเส้นสแกนในแต่ละกล้อง ได้ผลออกมาดังแสดงในตารางที่ 5.4

	กล้อง 1	กล้อง 2	กล้อง 3	กล้อง 4	กล้อง 5	หน่วย
D_{max}	100.2	100.4	100.2	100.2	100.6	ม.ม
L_{max}	501	502	501	501	503	(เส้น)
θ_1	86.38	84.88	86.38	86.38	83.74	องศา
θ_2	3.62	5.12	3.62	3.62	6.26	องศา
เอียง	ซ้าย	ซ้าย	ขวา	ซ้าย	ซ้าย	-

ตารางที่ 5.4 แสดงข้อมูลจากการหาค่าความเอียงของเส้นสแกน

ค่าความเอียงที่ได้นี้มีประโยชน์อย่างมาก ในการนำกลับมาแก้ไขความผิดพลาดที่เกิดขึ้น ซึ่งสามารถเลือกทำทางฮาร์ดแวร์ คือ การปรับแต่งตำแหน่งการวางของตัวอิมเมจเซ็นเซอร์ หรือทำทางด้านซอฟต์แวร์โดยการนำค่าความเอียงที่หาได้ไปชดเชยค่าที่ผิดเพี้ยนไป แต่อย่างไรก็ตามค่าความยาวเส้นผ่าศูนย์กลาง D_{max} นั้นไม่สามารถเชื่อถือได้ เพราะการเคลื่อนที่ของหัวอ่านมีอาการสั่นค่าที่ได้จากการอ่านอาจคลาดเคลื่อน ดังนั้นจึงควรแก้การสั่นก่อนเพื่อให้วัดค่าได้แน่นอน และอีกอย่างค่าความยาวเส้นผ่าศูนย์กลาง D_{max} มีความละเอียดเพียง 0.2 มิลลิเมตร เท่านั้น จึงทำให้ค่ามุมเอียงที่คำนวนได้หยาบมากเกินกว่าที่จะนำวิธีการคำนวนแบบนี้มาใช้งานจริง

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 6

สรุปและข้อเสนอแนะ

6.1 สรุป

เครื่องอ่านภาพแผ่นโลหะเรียบที่ออกแบบสร้างขึ้นนี้ ทำเพื่อการใช้งานเฉพาะกับโรงงาน ที่มีการเจาะประกอบ ซึ่งต้องการเน้นในการตรวจวัดความคลาดเคลื่อนตำแหน่งของรูเจาะ ดังนั้นจึงไม่เหมาะที่จะนำไปใช้กับงานอย่างอื่นที่มีความซับซ้อน โดยมีองค์ประกอบ และลักษณะการทำงานของเครื่องอ่านภาพแผ่นโลหะเรียบที่สร้างขึ้น พอสรุปเป็นข้อๆ ดังต่อไปนี้

1) เครื่องอ่านภาพแผ่นโลหะเรียบที่สร้างขึ้นนี้ ใช้การออกแบบระดับบอร์ด โดยใช้บอร์ดไมโครคอนโทรลเลอร์ 8 บิทเป็นตัวควบคุมระบบ การเชื่อมต่อระหว่างบอร์ดควบคุมและส่วนอื่นๆ ทำได้โดยผ่านพอร์ต อินพุต/เอาต์พุต ซึ่งมีอยู่ถึง 40 พอร์ต ส่วนที่ใช้กับกล้อง 5 ตัว นั้นออกแบบให้เป็นแบบการ์ดเสียบเพื่อให้ง่ายต่อการซ่อมแซม โดยการทำชุดเสริมขึ้นให้สามารถนำมาเสียบแทนที่ได้ทันที

2) การทำงานของเครื่องอ่านภาพแผ่นโลหะเรียบ สามารถครอบคลุมเนื้อที่กว้าง 1เมตร และยาวถึง 2.4 เมตร โดยมีการเคลื่อนที่ 0.2 มิลลิเมตรในการอ่านภาพแต่ละครั้ง ส่วนกลไกที่ใช้ขับเคลื่อน จะมีชุด linear motion ซึ่งประกอบด้วย lead screw shaft,ball screw nut,ball slide,ball slide shaft และ ball bearing โดยใช้สเต็ปปิ้งมอเตอร์เป็นตัวขับเคลื่อน lead screw ในการควบคุมแบบวงรอบเปิด

3) การอ่านภาพของเครื่องอ่านภาพแผ่นโลหะเรียบ มีความละเอียดถึง 5,000 จุด ต่อกล้อง โดยกล้องแต่ละตัวจับภาพกว้างประมาณ 250 มิลลิเมตร ดังนั้นทำให้ความหนาแน่นของจุดภาพที่ได้ประมาณ 508 dpi โดยแต่ละกล้องสามารถเก็บข้อมูลได้ 64 จุด และผลที่ได้จากการสแกนจะแยกแต่ละกล้องออกจากกัน ดังนั้นจึงมีเอาต์พุต ไฟล์ จากการสแกนถึง 5 ไฟล์

4) โปรแกรมการแก้ไขภาพและต่อภาพ เป็นส่วนที่ใช้สำหรับการแก้ไขความบกพร่องทางกลไก ตำแหน่งการติดตั้งของเลนส์ และตัวอิมเมจเซ็นเซอร์ ซึ่งมีผลต่อ ขนาดของภาพ ตำแหน่งของเส้นสแกน และการเอียงของเส้นสแกน ส่วนที่สามารถ แก้ไขได้ ก็คือ อัตราการขยายภาพ และการเลื่อนแถวของเส้นสแกน โดยสามารถปรับอัตราการขยายของกล้องให้ถูกต้อง และสามารถเลื่อนแถวของเส้นสแกนของกล้องทุกตัวให้มาอยู่ในแนวเส้นเดียวกัน หลังจากการปรับแต่งค่าในไฟล์ SYSTEM.CFG ที่นำมาใช้เป็นค่าคงที่ให้กับโปรแกรมนี้แล้ว จึงทำการทดสอบการสแกน ผลปรากฏว่ามีขนาดผิดเพี้ยนไปบ้าง เนื่องจากการติดตั้งตำแหน่งของลิเนียร์อิมเมจเซ็นเซอร์เอียงจากเส้นแนวระนาบ

5) ผลการทดสอบอ่านแผ่นโลหะเรียบ ความหนาขนาด 1 มม. พบว่าเครื่องอ่านภาพด้วยความเร็วประมาณ 5 นาทีต่อความยาว 1 เมตร (จากค่าเฉลี่ยในตารางที่ 5.1 ที่มีค่าประมาณ 1:30 นาทีต่อระยะการสแกน 300 มม.) โดยมีความกว้างการสแกน 1 เมตร ความยาวสูงสุด 2.4 เมตร ความแม่นยำ 0.6 มม. และ ความละเอียด 508 จุดต่อนิ้ว

6.2 ปัญหาและข้อเสนอแนะ

เครื่องอ่านภาพแผ่นโลหะเรียบที่สร้างขึ้นนี้เป็นเครื่องต้นแบบ ยังไม่สมบูรณ์ และ มีปัญหาที่ยังไม่ได้แก้ไขเนื่องจากเวลาที่จำกัด ซึ่งปัญหามีดังต่อไปนี้

1) การวางตำแหน่งของเลนส์ และตัวอิมเมจเซ็นเซอร์นั้น ในทางปฏิบัติไม่สามารถวางให้อยู่ในตำแหน่งที่ถูกต้องได้ ทำให้ความผิดพลาดหลายอย่างเกิดขึ้น เช่น อัตราการขยายและการที่ตำแหน่งเส้นสแกนของกล้อง 5 ตัว ไม่ตรงกัน ซึ่งก็สามารถแก้ไขได้ แต่มีอีกเรื่องที่แก้ไขได้ยาก คือ การเอียงของการติดตั้งอิมเมจเซ็นเซอร์ ซึ่งมีผลทำให้ความแม่นยำของการวัดขนาดผิดไป ดังนั้นการติดตั้งตัวอิมเมจเซ็นเซอร์ควรหาเครื่องมือที่มีความละเอียดแม่นยำสูงมาช่วยในการติดตั้ง จึงจะทำให้ปัญหานี้หมดไป

2) ปัญหาจากความเพี้ยนของเลนส์ ซึ่งทำให้การอ่านระยะจากภาพที่ได้ผิดไป เลนส์ที่ใช้เป็นเลนส์ชนิดไมโครใช้ถ่ายใกล้ ยังไม่สามารถหาข้อมูลเกี่ยวกับความเพี้ยนได้ การเลือกเลนส์ที่ให้ความเพี้ยนต่ำ ถ่ายได้ใกล้ และองศาจับภาพกว้างจะช่วยทำให้ความแม่นยำในการวัดดีขึ้น

3) ปัญหาอีกอย่างก็คือ หลักการทำงานของเครื่องที่สร้างขึ้นนี้ ต้องอาศัยความความเข้มแสง แยกเอาข้อมูลเฉพาะตรงขอบของแผ่นโลหะ แต่ใน

ทางปฏิบัติสีบนแผ่นโลหะมีสีไม่สม่ำเสมอ ขอบของภาพจิงไม่ชัดเจนและรอยเปื้อนบนแผ่นโลหะทำให้เกิดข้อมูลเทียม ทำให้ข้อมูลที่ได้จากการสแกนมีจำนวนข้อมูลมากเกินกว่าที่หน่วยความจำ FIFO จะสามารถเก็บได้ ดังนั้นในการอ่านภาพแผ่นโลหะเรียบจำต้องพ่นผิวของโลหะแผ่นเรียบด้วยสีขาวก่อน เพื่อให้เกิดความแตกต่างระหว่างแผ่นโลหะกับพื้นโต๊ะของเครื่องอ่านภาพมากยิ่งขึ้น การพ่นสีขาวบนผิวโลหะที่จะวัดในทางปฏิบัติเป็นไปได้ยาก และ ไม่สะดวก การแก้ไขในจุดนี้อาจทำได้โดยติดตั้งหลอดฟลูออเรสเซ็นต์อยู่ใต้โต๊ะและแผ่นโลหะ เป็นการเปลี่ยนทิศทางของแสง โดยการอ่านภาพแบบย้อนแสงซึ่งให้คอนทราสต์มากกว่าแบบอ่านภาพจากแสงสะท้อน แต่จะต้องเปลี่ยนให้แสงมาจากทางใต้โต๊ะของเครื่องอ่านภาพแผ่นโลหะเรียบ และ เปลี่ยนผิวโต๊ะจากวัตถุทึบแสงมาเป็นแผ่นกระจก เพื่อให้แสงจากใต้โต๊ะสามารถผ่านขึ้นมายังส่วนหัวอ่านได้

4) ในการออกแบบสร้างส่วนกลไกของเครื่องอ่านภาพแผ่นโลหะเรียบนี้ เนื่องจากเป็นเครื่องต้นแบบ การออกแบบยังมีข้อผิดพลาดบางอย่างเกี่ยวกับความแข็งแรงของโครงสร้างไม่เพียงพอ เมื่อมีการเคลื่อนไหวจึงทำให้เกิดการสั่นของหัวอ่าน ดังนั้นจึงส่งผลให้ข้อมูลที่ได้จากการสแกนมีความคลาดเคลื่อน ซึ่งปัญหาอันนี้สามารถแก้ไขได้ โดยการขยายช่วงกว้างการติดตั้งรางนำร่อง (ball slide shaft)ทั้งสองข้างของส่วนหัวอ่านให้กว้างขึ้น แต่ก็อยู่ในขอบเขตที่ไม่ทำให้ส่วนที่เคลื่อนไหวมีน้ำหนักมากเกินไป เพื่อไม่ให้สเต็บปิ้งมอเตอร์รับโหลดมากเกินไป

5) การสแกนมีความเร็วไม่มากนักเพียง 0.2 เมตรต่อนาที เป็นเพราะการขับเคลื่อนหัวอ่านนั้นใช้สเต็บปิ้งมอเตอร์ ซึ่งมีการทำงานไม่เร็วนัก ทางแก้อาจต้องเปลี่ยนมาใช้เซอร์โวมอเตอร์ (Servo motor) ที่ทำงานเร็วกว่า นอกจากนั้นความเร็วยังถูกจำกัดด้วยปัญหาการเก็บข้อมูลลงฮาร์ดดิสก์ทางด้านไมโครคอมพิวเตอร์ ซึ่งจะต้องเก็บข้อมูลเข้าดิสก์ทุกๆเส้นสแกนซึ่งทำให้การทำงานช้าลง ปัญหานี้สามารถแก้ไขโดยการเพิ่มหน่วยความจำเพื่อนำมาทำแรมดิสก์บนเครื่องไมโครคอมพิวเตอร์ เพราะการเก็บข้อมูลเข้าแรมดิสก์เร็วกว่าฮาร์ดดิสก์มาก

6) ปัญหาสุดท้ายที่จะกล่าวก็คือ ระบบกลไก เนื่องจากความยาวของโต๊ะสแกนที่ยาวถึง 3 เมตร เพื่อให้ได้ระยะสแกน 2.4 เมตร ประกอบกับ lead screw shaft มีขนาดเส้นผ่าศูนย์กลางเพียง 1 นิ้วเท่านั้น มีโค้งงอ

เกิดขึ้นทำให้มีแรงเสียดทานมากขึ้น อีกทั้งหัวอ่านมีน้ำหนักประมาณ 100 กิโลกรัม ซึ่งมีผลทั้งต่อกำลังการขับเคลื่อนที่ต้องใช้มากขึ้น และยังทำให้สเต็บปิ้งมอเตอร์ไม่สามารถใช้งานที่ความเร็วสูง ทั้งนี้เพราะมีแรงบิดต่ำเกินกว่าที่จะนำมาใช้งานได้ ต้องเปลี่ยน lead screw shaft ให้มีขนาดใหญ่ขึ้นเป็น 2 นิ้ว เพื่อไม่ให้เกิดการโค้งงอของ shaft หรือไม่ก็ต้องลดขนาดความยาวของโต๊ะลงมาเหลือเพียง 2 เมตร จึงจะไม่ทำให้เกิดการโค้งงอของแกนได้

7) ในการวิจัยนี้ได้นำลิเนียร์อิมเมจเซ็นเซอร์ชนิดเลนส์มาใช้ ซึ่งได้รับผลกระทบทำให้เกิดการคลาดเคลื่อนขึ้นจากการใช้เลนส์ ไม่ว่าจะเป็นความผิดเพี้ยนจากเลนส์ ความสูงของหัวอ่าน และ ปริมาณการใช้แสง ดังนั้นยังมีทางเลือกอีกทาง คือ การใช้ลิเนียร์อิมเมจเซ็นเซอร์ชนิดคอนแท็ค ซึ่งสามารถลดปัญหาดังที่ได้กล่าวมาแล้ว และ สามารถใช้แทนกันได้โดยไม่ต้องเปลี่ยนแปลงวงจร แต่อย่างไรก็ตามลิเนียร์อิมเมจเซ็นเซอร์ชนิดคอนแท็คไม่มีผลิตอยู่ในปัจจุบัน จะเป็นชนิดที่ใช้ในเครื่องโทรสาร มีความยาว 250 มม. เพื่อนำมาใช้ในการสแกนอ่านแผ่นโลหะกว้าง 1 เมตร ก็จำเป็นต้องนำมาต่อเหลื่อมกันหลายตัว การต่อเหลื่อมกันจะทำให้ต้องทำการต่อภาพเช่นเดียวกับกรณีการใช้กล้อง ทำให้มีปัญหาในการต่อภาพเช่นกัน

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

เอกสารอ้างอิง

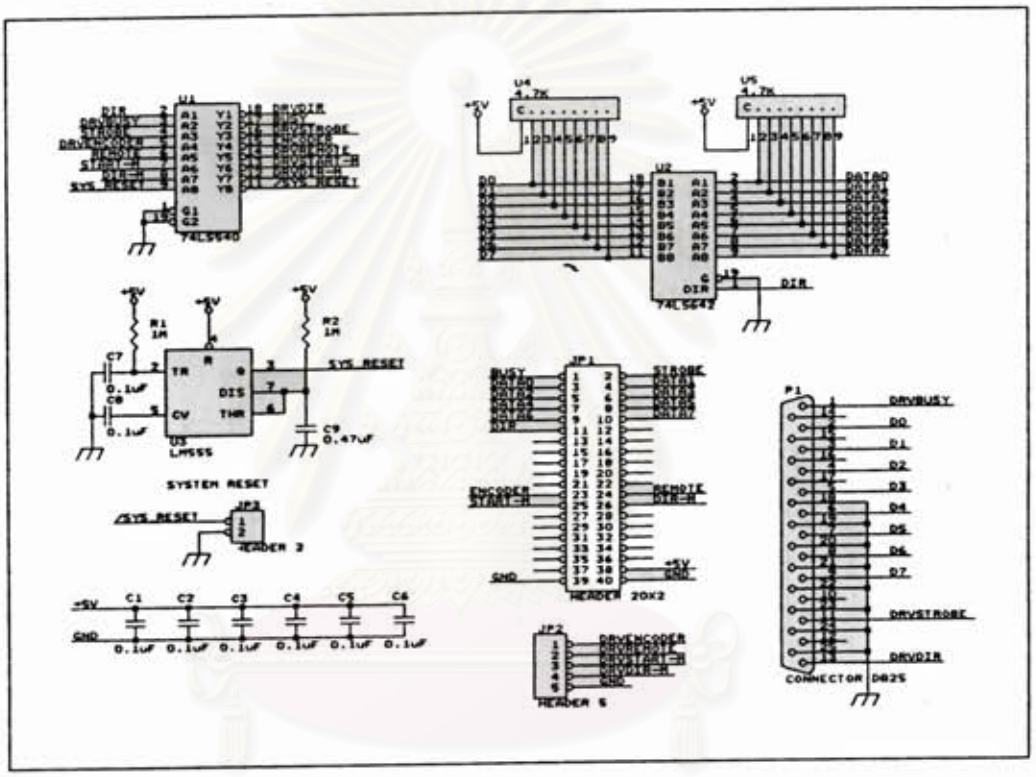
1. Charles Easteal,Gordon Davies, Soft Ware Engineering: Analysis and Design , McGRAW-HILL,1989.
2. Alan D. Wilcox, Engineering Design for Electrical Engineers,Prentice-Hall,1990.
3. Oce Graphics CO.,LTD,G6000 Series:Professional monochrome AO scanner , CATALOG :Engl.Exp.-05/92-EPCOM
4. Fujitsu LIMITED, revised contents : Advance Release. image scanner ,specification No.50FH6809E-01. FAI copyright,August,1990.
5. Harry H. Poole, fundamental of Robotics Engineering. 1989.
6. "หลักการทำงานของ CCD IMAGE SENSOR",เซมิฯ,ฉบับที่ 88 (ก.ย.-ต.ค.2531).
7. G.S. Hopson , Charge-Transfer Devices ,Edward Arnold-(Publishers) Ltd.,1978.
8. M.J.Howes,D.V. Morgan,Charge-Coupled Devices and System,John Wiley & Sons Ltd.,1979.
9. ANA-DIGIT CO.,LTD,DT2856 Line Scan, ANA-DIGIT PRODUCT CATALOG:DATA TRANSLATION,1992-1993.
10. "ทางยาวโฟกัส และองศาการรับภาพ",PHOTOTECH,ฉบับที่ 6 (2535). หน้า 99-103.
11. Warren J. Smith,Modern Optical Engineering:The Design of Optical Systems, McGRAW-HILL,1991.
12. THK CO.,Ltd.,THK LM SYSTEM-Ball Srews ,CATALOG.No.75-1BE,THK CO.,Ltd.
13. THK CO.,Ltd., THK LM SYSTEM-Linear Motion Systems. CATALOG.No.100-1AE ,Tokyo,Japan,THK CO.,Ltd.
14. Parker Hannifin Corporation , Microstepping Systems-OEM650 Series:Preliminary,Bulletin840030M9/91.
15. National/Panasonic,Specification of MN3662 and MN3664 :Image Sensor,1989.
16. ETT จำกัด,CP-Z84C11: PARALLEL I/O CONTROLLER,2534.

ภาคผนวก

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

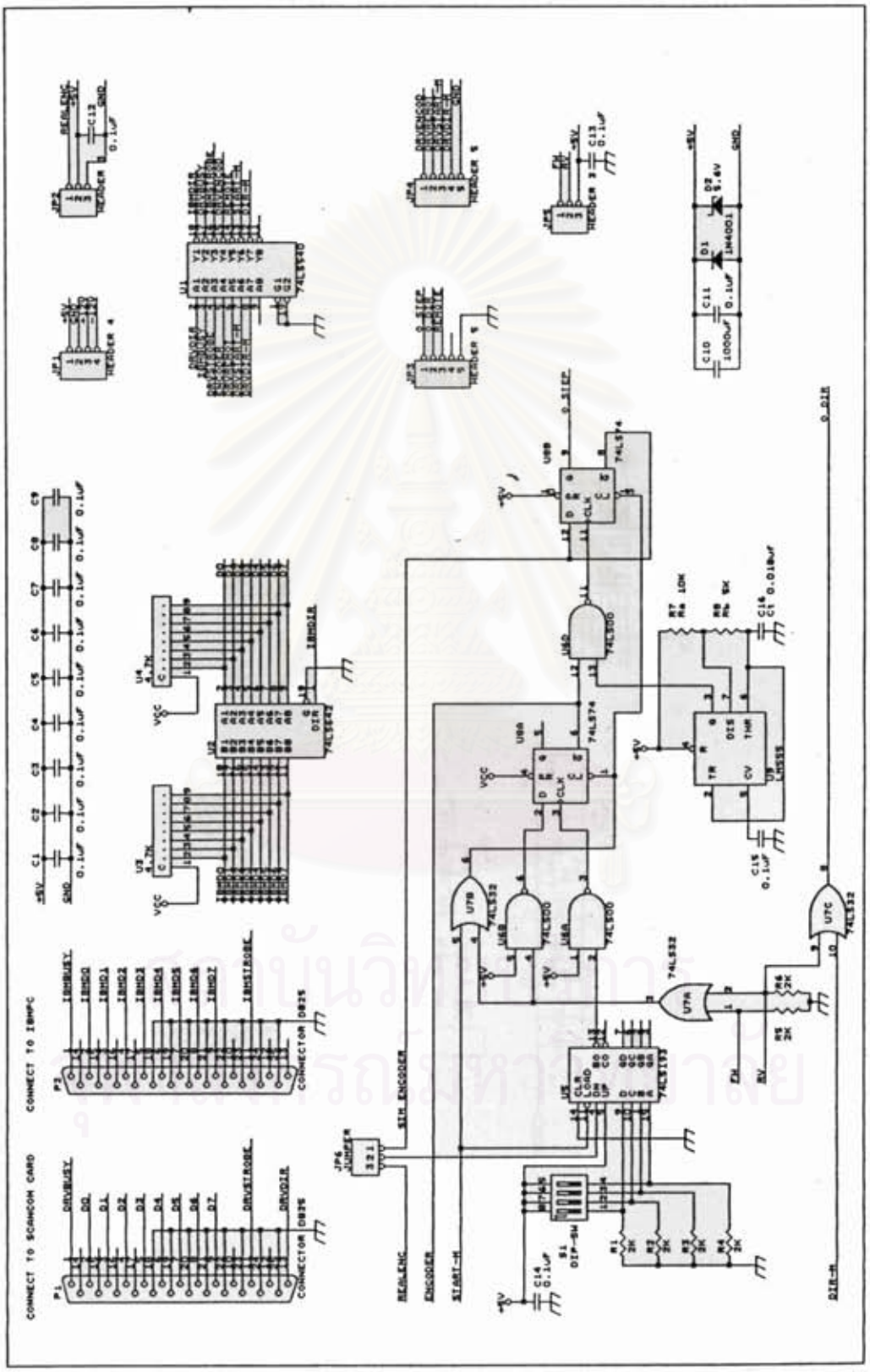
ภาคผนวก ก

วงจรอิเล็กทรอนิกส์ที่ใช้ในเครื่องอ่านภาพแผ่นโลหะเรียบ

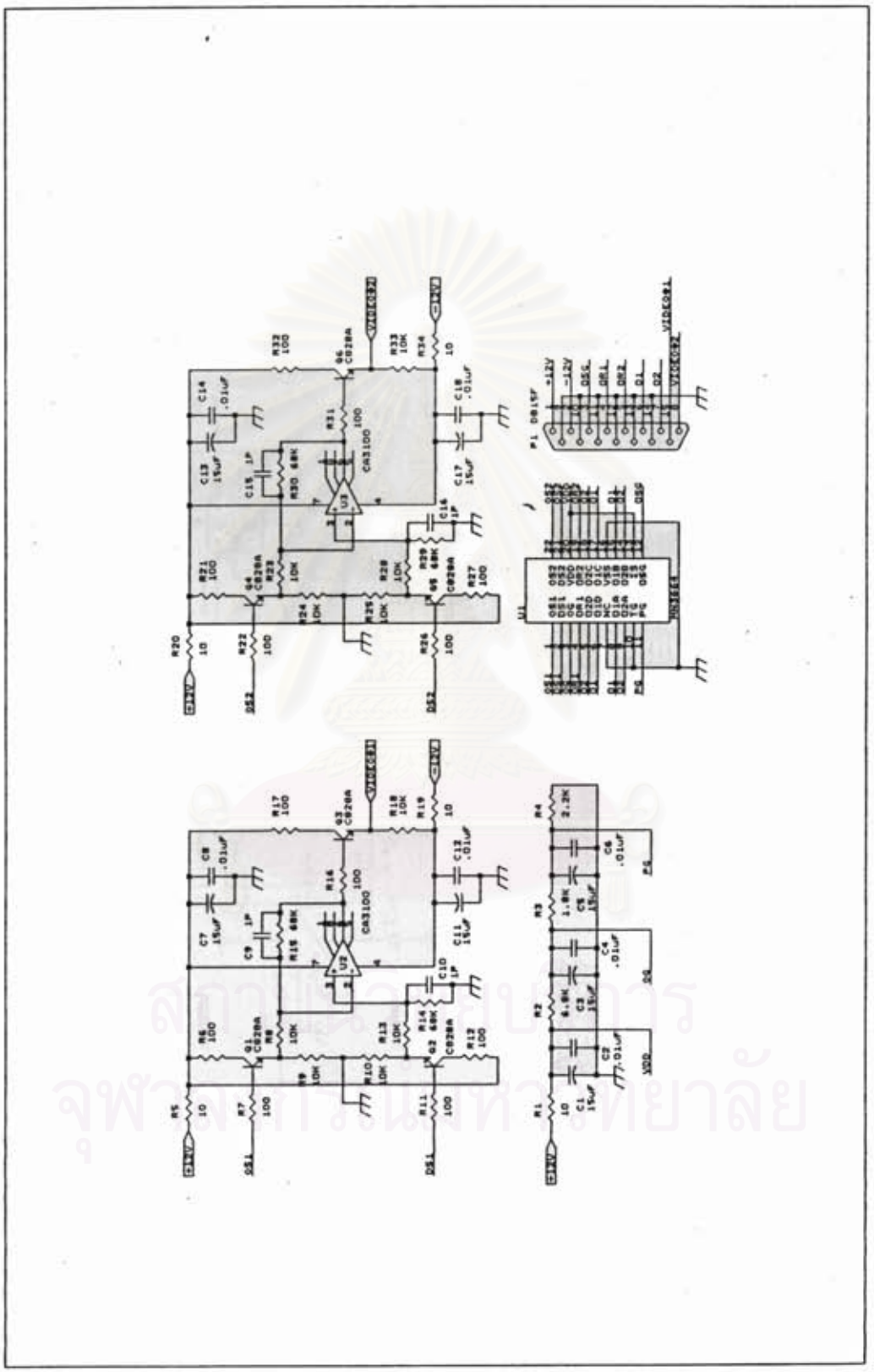


วงจรการ์ดสื่อสารข้อมูล

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

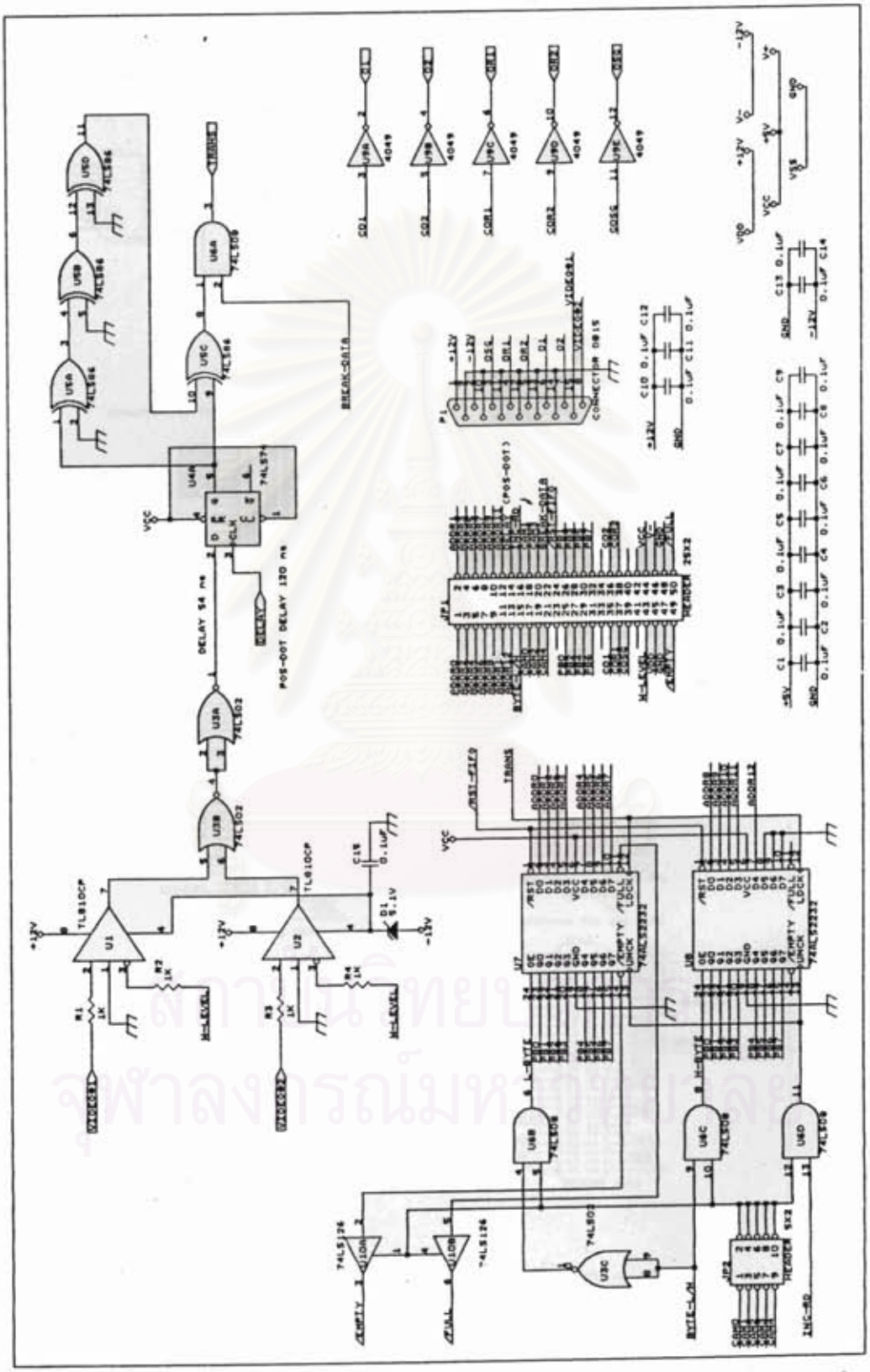


วงจรบอร์ดอินเตอร์เฟส

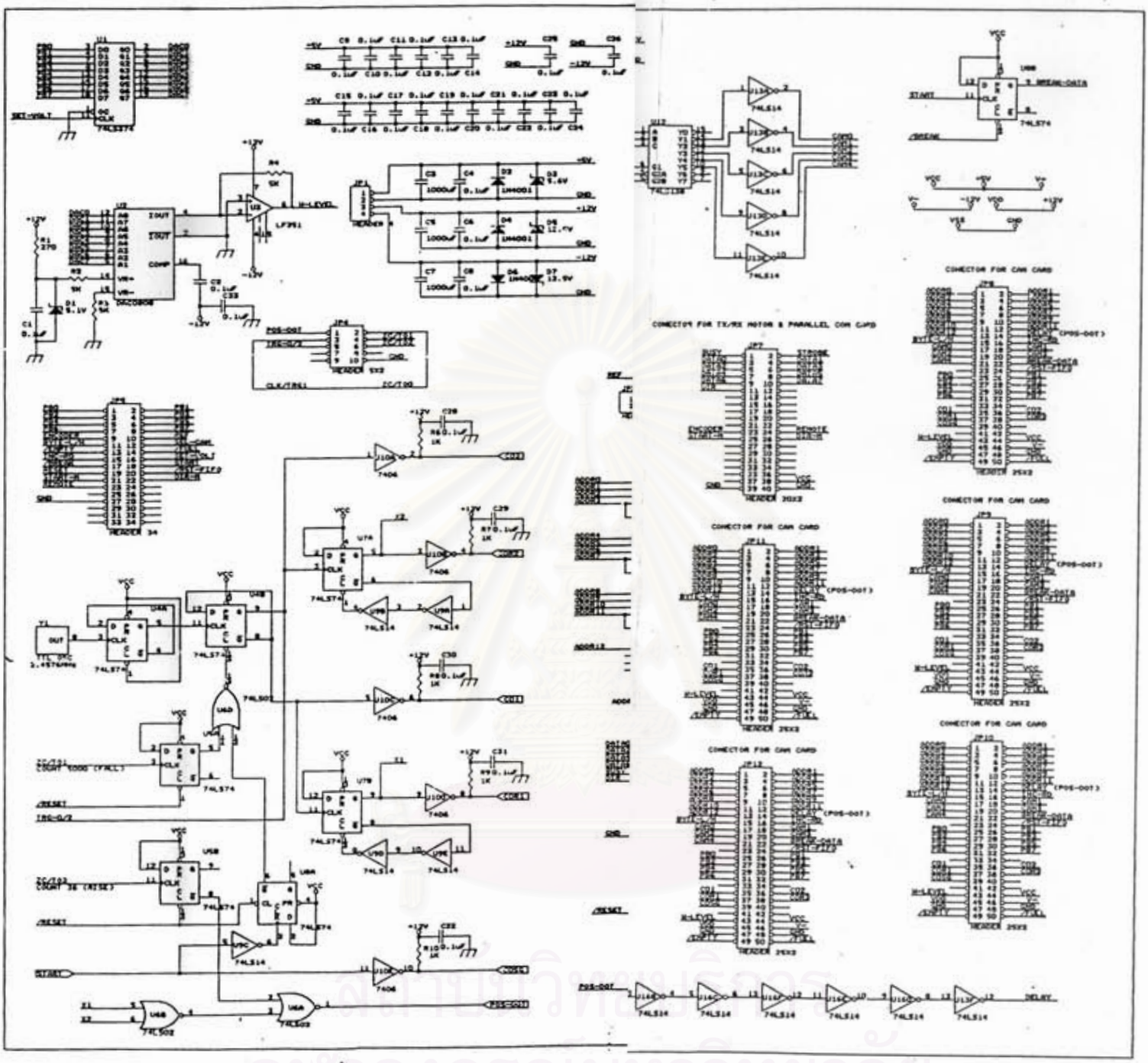


วงจรบอร์ดขยายสัญญาณภาพ

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย



วงจรการ์ดควบคุมกล้อง



สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ภาคผนวก ข

โปรแกรมควบคุมการทำงาน

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

File SYSTEM.CFG

```
Xc1=1.0315
Xc2=1.0315
Xc3=1.0355
Xc4=1.0380
Xc5=1.0385
Yshift1=8
Yshift2=2
Yshift3=0
Yshift4=3
Yshift5=3
Ofront1=491
Ofront2=491
Ofront3=509
Ofront4=480
Ofront5=494
```

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ไฟล์โปรแกรมที่มีอยู่ในระบบ

1) IBMSCAN.C เป็นโปรแกรมควบคุมการทำงานของเครื่องไมโครคอมพิวเตอร์ (หน้าที่ 91)

2) ADDFILE.C เป็นโปรแกรมรวมไฟล์ 5 ไฟล์ ที่ได้จากสแกนภาพแผ่นโลหะ (หน้าที่ 106)

3) SCANSYS.C เป็นโปรแกรมควบคุมการทำงานของเครื่องอ่านภาพแผ่นโลหะเรียบ (หน้าที่ 113)

หมายเหตุ

(1) และ (2) เขียนด้วยภาษา C คอมไพล์ด้วย Turbo C 2.0 เป็นโปรแกรมที่ใช้บนเครื่อง IBM PC

(3) เขียนด้วยภาษา C คอมไพล์ด้วย Z World - C และ เมื่อได้ไฟล์ SCANSYS.HEX แล้วทำการแปลงเป็น Binary file ชื่อ SCANSYS.BIN จากนั้นจึงนำไฟล์ที่ได้มาโปรแกรมอีพรอมเบอร์ 27C256 โดยใช้เครื่องโปรแกรมยี่ห้อ Xeltex ซึ่งโปรแกรมที่อยู่ในอีพรอมนั้นใช้ควบคุมการทำงานของซีพียูเบอร์ Z84C11 บนบอร์ดไมโครคอนโทรลเลอร์

```
#include <stdio.h>
#include <dos.h>
#include <string.h>
#include <graphics.h>
#include <stdlib.h>
#include <conio.h>
#include <alloc.h>

#define  ZERO            0x00
#define  BUSY            0x80
#define  STROBE          0xFF
#define  NOT_STROBE      0x00
#define  END_OF_COMMAND  0x00
#define  END_OF_MASSAGE  0xFF
#define  END_OF_STATUS   0xFF
#define  COMMAND1        "scan"
#define  COMMAND2        "threshold"
#define  COMMAND3        "ali&cal"
#define  COMMAND4        "begin"
#define  X_MIN           69
#define  Y_MIN           199
#define  Y_MAX           179
#define  X_AXS           209
#define  TEN_CM          10
#define  ONE_M           100
#define  POINT_MAX       5000
      /* COLOR */
#define  EGA_BLACK         0
#define  EGA_BULE          1
#define  EGA_GREEN         2
#define  EGA_CYAN          3
#define  EGA_RED           4
#define  EGA_MAGENTA       5
#define  EGA_LIGHTGREY     7
#define  EGA_BROWN         20
#define  EGA_DARKGREY      56
#define  EGA_LIGHTBULE     57
#define  EGA_LIGHTGREEN    58
#define  EGA_LIGHTCYAN     59
#define  EGA_LIGHTRED      60
#define  EGA_LIGHTMAGENTA  61
#define  EGA_YELLOW        62
#define  EGA_WHITE         63
      /* KEYBOARD CODE */
```

```
#define  LFARROW          75
#define  RTARROW          77
#define  UPARROW          72
#define  DNARROW          80
      /* draw key */
#define  KEY_XWIDTH       20
#define  KEY_YWIDTH       20
#define  KEY1_XADDR       535      /* left arrow */
#define  KEY1_YADDR       300
#define  KEY2_XADDR       585      /* right arrow */
#define  KEY2_YADDR       300
#define  KEY3_XADDR       560      /* up arrow */
#define  KEY3_YADDR       275
#define  KEY4_XADDR       560      /* down arrow */
#define  KEY4_YADDR       300
      /* function */
find_port();
tx_data();
rx_data();
ali_n_cal();
graph();
rule();
helpwindow();
menu();
page_menu();
confirm_run();
confirm_exit();
titlewindow();

FILE  *fp1,*fp2,*fp3,*fp4,*fp5;
unsigned int   data[5][40],index[5];
int   extend = 100,value = 100,long_size;
unsigned int   data_port,strobe_port,busy_port;
unsigned char  command[14],massage[5][130],status[5][2];
unsigned char  cam=0;

int   GraphDriver = VGA,GraphMode = VGAMED;
int   position=0;

main() {

   clrscr();
   find_port();
   initgraph(&GraphDriver,&GraphMode,"");
```

```
    cleardevice();
    menu();
    closegraph();
}

find_port() {

  int  addr_prn;

  addr_prn = peek(0x40,0x08);          /* addr printer port in bios */
  data_port = addr_prn;
  busy_port = addr_prn + 1;
  strobe_port = addr_prn + 2;
  outportb(strobe_port,STROBE);                    /* give LO */
}

tx_data() {
   char  i;

   i=0;
   do {
     while ( (inportb(busy_port) & BUSY) > NULL );     /* HI pass */
     outportb(data_port,command[i]);                    /* write data */
     outportb(strobe_port,NOT_STROBE);                  /* give HI */
     while ( (inportb(busy_port) & BUSY) == NULL );    /* LO pass */
     outportb(strobe_port,STROBE);                      /* give LO */
     i++;
   } while (command[i-1] != END_OF_COMMAND);
   outportb(data_port,0xFF); /* hold up */
}

rx_data() {
   unsigned char   i,toggle,sel_cam;

   index[0]=index[1]=index[2]=index[3]=index[4]=0; /* count amont of data *
for (sel_cam=0;sel_cam<5;sel_cam++) {
   i=0;toggle = ZERO;
   /* RX DATA */
   do {
     toggle = ~toggle;
     outportb(strobe_port,NOT_STROBE);                  /* give HI */
     while ( (inportb(busy_port) & BUSY) >  NULL );     /* HI pass */
     massage[sel_cam][i] = inportb(data_port);         /* read data */
     outportb(strobe_port,STROBE);                      /* give LO */
```

```
      while ( (inportb(busy_port) & BUSY) == NULL );     /* LO pass */
      i++;
      if (toggle == ZERO) {
  data[sel_cam][i/2-1] = (((unsigned int)massage[sel_cam][i-2] << 8)|
 (unsigned int)massage[sel_cam][i-1] );
  data[sel_cam][i/2] = ZERO;
          index[sel_cam]++;
      }
    } while ( (massage[sel_cam][i-1] & toggle) != END_OF_MASSAGE );
    /* RX STATUS */
    i=0;
    do {
      outportb(strobe_port,NOT_STROBE);                  /* give HI */
      while ( (inportb(busy_port) & BUSY) >  NULL );     /* HI pass */
      status[sel_cam][i] = inportb(data_port);          /* read data */
      outportb(strobe_port,STROBE);                      /* give LO */
      while ( (inportb(busy_port) & BUSY) == NULL );     /* LO pass */
      i++;
    } while (status[sel_cam][i-1] != END_OF_STATUS);
}
}

ali_n_cal() {

  char   key = NULL;
  char   drawpage = ZERO;
  char   value_str[4];
  int    page=0;

  setvisualpage(page);
  setactivepage(1-page);      /* set active page for write */
  do {
    setfillstyle(1,EGA_LIGHTGREY);
    bar(0,0,639,349);                /* external frame */
    setfillstyle(1,EGA_BLUE);
    bar(9,9,630,340);                /* internal frame */
    setfillstyle(1,EGA_LIGHTCYAN);
    bar(24,94,614,244);              /* external frame display */
    setcolor(EGA_LIGHTCYAN);
    settextstyle(0,0,2);
    outtextxy(X_MIN,40,"Metal Sheet Image Scanner System");
    settextstyle(2,0,7);
    setfillstyle(1,EGA_DARKGREY);
    bar(29,99,609,239);              /* frame display */
```

```
      rule();
      setcolor(EGA_LIGHTGREEN);
      if (extend == ONE_M)  outtextxy(X_MIN+235,Y_MIN+29,"(cm)");
      if (extend == TEN_CM) outtextxy(X_MIN+235,Y_MIN+29,"(mm)");
      /* key command */
      /* left */
      setcolor(EGA_LIGHTRED); outtextxy(X_MIN+50,259,"ZOOM");
      setcolor(EGA_WHITE);outtextxy(X_MIN+87,259,"I");
      setcolor(EGA_LIGHTRED);outtextxy(X_MIN+93,259,"N");
      setcolor(EGA_LIGHTRED);outtextxy(X_MIN+50,279,"ZOOM");
      setcolor(EGA_WHITE);outtextxy(X_MIN+87,279,"O");
      setcolor(EGA_LIGHTRED);outtextxy(X_MIN+95,279,"UT");
      setcolor(EGA_LIGHTRED);outtextxy(X_MIN+50,319,"E");
      setcolor(EGA_WHITE);outtextxy(X_MIN+59,319,"X");
      setcolor(EGA_LIGHTRED);outtextxy(X_MIN+66,319,"IT");
      /* right */
      setcolor(EGA_LIGHTRED); outtextxy(X_MIN+300,259,"CAM");
      setcolor(EGA_WHITE);outtextxy(X_MIN+327,259,"1");
      setcolor(EGA_LIGHTRED); outtextxy(X_MIN+300,274,"CAM");
      setcolor(EGA_WHITE);outtextxy(X_MIN+327,274,"2");
      setcolor(EGA_LIGHTRED); outtextxy(X_MIN+300,289,"CAM");
      setcolor(EGA_WHITE);outtextxy(X_MIN+327,289,"3");
      setcolor(EGA_LIGHTRED); outtextxy(X_MIN+300,304,"CAM");
      setcolor(EGA_WHITE);outtextxy(X_MIN+327,304,"4");
      setcolor(EGA_LIGHTRED); outtextxy(X_MIN+300,319,"CAM");
      setcolor(EGA_WHITE);outtextxy(X_MIN+327,319,"5");
      /**************************/
      setvisualpage(1-page);
      setactivepage(page);      /* set active page for write */
      drawpage = ~drawpage;
   } while (drawpage != ZERO);
   while ( (strcmp(command,COMMAND3) == 0) && (key != 'x') ) {
      setactivepage(1-page);    /* set active page for write */
      rx_data();
      /******* key board management *******/
      key = NULL;                        /* clear keyboard */
      if ( kbhit() != 0 )   key = getch();
      if (key != NULL) {
if ( key == 'o')  extend = 100;    /* zoom out */
if ( extend == 100 ) position = 0; /* zoom out not select section */
if ( key == 'i')  {
   extend = 10;                    /* zoom in */
   position = 0;
}
```

```
if ( key == '1')   cam = 0;
if ( key == '2')   cam = 1;
if ( key == '3')   cam = 2;
if ( key == '4')   cam = 3;
if ( key == '5')   cam = 4;
if ( extend == 10 ) {
   if ( (key == LFARROW) && (position > 0) )    position -= 100;
   if ( (key == RTARROW) && (position < 900) )  position += 100;
}
if ( (key == UPARROW) && (value < 255) ) {
   value+= 1;
   strcpy(command,"threshold/");
   command[12] = value%10+'0';     /* add ZERO lead char */
   command[11] = ((value%100)-(value%10))/10+'0';
   command[10] = ((value%1000)-(value%100))/100+'0';
   command[13] = NULL;
   tx_data();
}
if ( (key == DNARROW) && (value > 0) ) {
   value-= 1;
   strcpy(command,"threshold/");
   command[12] = value%10+'0';     /* add ZERO lead char */
   command[11] = ((value%100)-(value%10))/10+'0';
   command[10] = ((value%1000)-(value%100))/100+'0';
   command[13] = NULL;
   tx_data();
}
    }
    if ( key != 'x' )   strcpy(command,"ali&cal");
    else   strcpy(command," ");          /* give scanner to stop */
    tx_data();
    setfillstyle(1,EGA_DARKGREY);
    bar(29,99,609,239);                  /* frame display */
    rule();
    setcolor(EGA_LIGHTGREEN);
    if (extend == ONE_M)  outtextxy(X_MIN+235,Y_MIN+29,"(cm)");
    if (extend == TEN_CM) outtextxy(X_MIN+235,Y_MIN+29,"(mm)");
    if (extend == 10)     outtextxy(375,104,"\n ZOOM");
    if (cam == 0)
      outtextxy(380,114,"CAM 1");
    if (cam == 1)
      outtextxy(380,114,"CAM 2");
    if (cam == 2)
      outtextxy(380,114,"CAM 3");
```

```
      if (cam == 3)
        outtextxy(380,114,"CAM 4");
      if (cam == 4)
        outtextxy(380,114,"CAM 5");
      setcolor(EGA_YELLOW);
      sprintf(value_str,"%d",value);        /* display threshold value */
      outtextxy(545,104,value_str);
      outtextxy(450,104,"\n THRESHOLD");
      /*  status of data buffer in scanner */
      setcolor(EGA_LIGHTRED);
      if ( (status[cam][0] == 0x66) || (status[cam][0] == 0x67)) {
setcolor(EGA_LIGHTCYAN);
if (status[cam][0] == 0x66)
  outtextxy(70,104,"\n STATUS : DATA ERROR");
if (status[cam][0] == 0x67)
  outtextxy(70,104,"\n STATUS : DATA OVER");
        setcolor(EGA_LIGHTRED);
      }
      else {
        if ((status[cam][0] & 0x30) == 0x10) {
  outtextxy(70,104,"\n STATUS : BUFFER FULL");
  outtextxy(70,114,"\n MEMORY NOT ENAUGH FOR WORK SHEET");
        }
        if ((status[cam][0] & 0x30) == 0x20) {
  outtextxy(70,104,"\n STATUS : BUFFER EMPTY");
  outtextxy(70,114,"\n CAN'T SEE OR NOT WORK SHEET");
        }
        if ((status[cam][0] & 0x30) == 0x30)  {
  outtextxy(70,104,"\n STATUS : DATA IN BUFFER");
  if (data[cam][1] != ZERO)  graph();
        }
      }
      /****** set visual page for display ********/
      setvisualpage(1-page);
      if (page==1) page=0;
      else page=1;

  }
  setactivepage(0);
  cleardevice();
}

graph() {
```

```
  char  i,number[7];
  int   xstart,ystart,xend,yend;
  char  toggle,over=0;

  i=0; toggle=ZERO;
  if (position > 0) {
    while  ( (data[cam][i] < (position)*5) && (data[cam][i] != NULL) ) {
i++; toggle = ~toggle;
    }
    if (i > 63)  over=1;
    if (over == 0) {
      xend = (10*data[cam][i])/extend+X_MIN-(position*5);
      if (xend > X_MIN+POINT_MAX/10)  {
 xend = X_MIN+POINT_MAX/10;
 over = 1;                  /* over range */
      }
      if (xend >= X_MIN) {
 if (toggle == ZERO)  ystart = yend = Y_MIN;
 else   ystart = yend = Y_MAX;
 setcolor(EGA_LIGHTBLUE);
 line(X_MIN,ystart,xend,yend);
      }
      else  over = 1;              /* under range */
    }
  }
  /* draw */
 if (over == 0) {
   setcolor(EGA_LIGHTMAGENTA);
   do {
     xstart = xend = (10*data[cam][i])/extend+X_MIN-(position*5);
     if (xstart < X_MIN+POINT_MAX/10) {
if (toggle==ZERO) { ystart = Y_MIN; yend = Y_MAX; }    /* edge */
else  { ystart = Y_MAX; yend = Y_MIN; }
line(xstart,ystart,xend,yend);
if (data[cam][i+1]>data[cam][i]) {              /* width value */
   if (extend == 10)
     sprintf(number,"%4.1f",((float)(data[cam][i+1]-data[cam][i]))
    /(2*extend) );
   else
     sprintf(number,"%4.2f",((float)(data[cam][i+1]-data[cam][i]))
    /(2*extend) );
   settextstyle(2,1,7);
   outtextxy(xstart+15,Y_MAX-50,number);
   settextstyle(2,0,7);
```

```
 }
 if (toggle==ZERO)  ystart = Y_MAX;
 else  ystart = Y_MIN;
 xend = (10*data[cam][i+1])/extend+X_MIN-(position*5);
 if (xend > X_MIN+POINT_MAX/10)  {
    xend = X_MIN+POINT_MAX/10;
    over = 1;
 }
 if (xend > 0)                                          /* line */
    line(xstart,ystart,xend,yend);
 if (over != 1)  {                                      /* not over */
    if (toggle==ZERO) { ystart = Y_MAX; yend = Y_MIN; }  /* edge */
    else  { ystart = Y_MIN; yend = Y_MAX; }
    xstart = xend;
    line(xstart,ystart,xend,yend);
 }
      }
      toggle = ~toggle;
      i++;
      xend = (10*data[cam][i])/extend+X_MIN-(position*5);
    } while ( (data[cam][i+1] != ZERO) && (data[cam][i+1] <= 5000)
      && (xend <= X_MIN+POINT_MAX/10));
 }
}

rule() {

   char  i,s[6],ten;
   int   xstart,ystart,xend,yend;

   xstart = X_MIN; xend = X_MIN+POINT_MAX/10;
   ystart = yend = X_AXS;
   setcolor(EGA_YELLOW);
   line(xstart,ystart,xend,yend);
   for (i=0,ten=0;i<=POINT_MAX/50;i++) {
     xstart = xend = X_MIN+(i*5);
     if (i==(TEN_CM*ten)) { ystart = X_AXS-4; yend = X_AXS+4;
  sprintf(s,"%3.1f",(float)(i+position)/4);
  setcolor(EGA_RED);
  outtextxy(xstart-7,X_AXS+6,s);
  ten++;
}
     else { ystart = X_AXS-2; yend = X_AXS+2; }
     setcolor(EGA_CYAN);
```

```
      line(xstart,ystart,xend,yend);
    }
}

helpwindow()  {

  /* help page 1 */
  page_menu();
  setcolor(EGA_WHITE);
  outtextxy(235,110,"*** HELP ***");
  outtextxy(50,130,"Welcome to the Metal Sheet Image Scanner System.  Before you run");
  outtextxy(50,140,"this system you must be turn on Scanner machine.  After,you can use")
  outtextxy(50,150,"facility of system.  ");
  outtextxy(50,170,"COMMAND LIST:");
  outtextxy(50,180,"scan                    use to start scanning");
  outtextxy(50,190,"alignment & calibrate   use to alignment and calibration");
  outtextxy(50,200,"begin                   use to move position of head-reader");
  outtextxy(50,220,"Press any key to continue");
  getch();
  page_menu();
  /* help page 2 */
  setcolor(EGA_WHITE);
  outtextxy(50,110,"COMMAND LIST:");
  outtextxy(50,130,"threshold               use to adjustly level of threshold");
  outtextxy(50,140,"zoom in                 use to see description of work sheet");
  outtextxy(50,150,"zoom out                use to see cover all of work sheet");
  outtextxy(50,160,"exit                    use to exit to main menu");
  outtextxy(50,170,"quit                    use to exit to DOS");
  outtextxy(50,180,"cam1..cam5              use for select cam in ali&cal mode");
  outtextxy(50,190,"");
  outtextxy(50,220,"Press any key to continue");
  getch();
  page_menu();
}

menu() {

    char  func_key;
    char  confirm,line_number[6],number[3];
    int i,j,convert;

  confirm_run();
  confirm = getch();
  while ( confirm == 'y')   {  /* confirm start and end system */
```

```
  helpwindow();
  strcpy(command,"threshold/100");          /* default value */
  tx_data();
  titlewindow();
  do  {
     func_key = NULL;
     if ( kbhit() != 0 )  func_key = getch();
     if ( func_key == 'a')  {  /* alignment and calibrate */
strcpy(command,"ali&cal");
tx_data();
ali_n_cal();
titlewindow();
     }
     /* move head to bigin */
     if ( func_key == 'b')  {
strcpy(command,"begin");
tx_data();
        setfillstyle(1,EGA_DARKGREY);
bar(29,99,609,239);              /* frame display */
outtextxy(50,120,"Please wait for set head to begin.....");
tx_data();
setcolor(EGA_DARKGREY);
outtextxy(50,120,"Please wait for set head to begin.....");
        setcolor(EGA_WHITE);
outtextxy(50,120,"Now,head reader at on starting point");
outtextxy(50,180,"Press any key to continue");
getch();
titlewindow();
     }
     /** start scanning **/
     if ( func_key == 's')  {
cleardevice();
page_menu();
outtextxy(50,120,"Please enter long size in CM unit (MAX 240 CM)");
j = 0;number[0] = number[1] = number[2] = '0';
do {
 number[j] = getch();
 if ( (number[j] >= '0') && (number[j] <= '9') ) {
   sprintf(line_number,"%c",number[j]);
   outtextxy(300+(j*9),150,line_number);
   j++;
   if (number[0] == '0') {
     setfillstyle(1,EGA_DARKGREY);
     bar(29,99,609,239);              /* frame display */
```

```
      outtextxy(50,120,"not enter zero lead other number");
      j = 0;
    }
    if (j > 3) {
      setfillstyle(1,EGA_DARKGREY);
      bar(29,99,609,239);              /* frame display */
      outtextxy(50,120,"Please enter long size 3 digit in CM unit");
      j = 0;
    }
  }
} while (number[j] != 0x0d); /* check key return */
number[j] = 0x00;
if (j == 1)
 long_size=(number[0]-'0');
if (j == 2)
 long_size=(number[0]-'0')*10+(number[1]-'0');
if (j == 3)
 long_size=(number[0]-'0')*100+(number[1]-'0')*10+(number[2]-'0');
if (long_size > 240)
 long_size = 0;                  /* MAX RANGE */
        setfillstyle(1,EGA_DARKGREY);
bar(29,99,609,239);               /* frame display */
strcpy(command,"scan/");
command[7] = long_size%10+'0';      /* add ZERO lead char */
command[6] = ((long_size%100)-(long_size%10))/10+'0';
command[5] = ((long_size%1000)-(long_size%100))/100+'0';
command[8] = NULL;
tx_data();
        setcolor(EGA_WHITE);
outtextxy(50,120,"Please wait for duration time of scanning.....");
convert = long_size * 50;
/* duration of scanning */
fp1 = fopen("cam1.scn","w");
fp2 = fopen("cam2.scn","w");
fp3 = fopen("cam3.scn","w");
fp4 = fopen("cam4.scn","w");
fp5 = fopen("cam5.scn","w");
for (j=0;j < convert;j++) {
  sprintf(line_number,"%3d CM",j*2/100+1);/* disp thresh value */
  setcolor(EGA_WHITE);
  ou textxy(300,150,line_number);
  rx_data();
  if (j == 0) {
     fprintf (fp1,"File_cam1.scn");   /* title file */
```

```
      fprintf (fp2,"File_cam2.scn");   /* title file */
      fprintf (fp3,"File_cam3.scn");   /* title file */
      fprintf (fp4,"File_cam4.scn");   /* title file */
      fprintf (fp5,"File_cam5.scn");   /* title file */
    }
    fprintf (fp1,"\nS %d",j+1);   /* seperate line */
    fprintf (fp2,"\nS %d",j+1);   /* seperate line */
    fprintf (fp3,"\nS %d",j+1);   /* seperate line */
    fprintf (fp4,"\nS %d",j+1);   /* seperate line */
    fprintf (fp5,"\nS %d",j+1);   /* seperate line */
    for (i=0;i<index[0];i++) {
      fprintf(fp1,"\n%d",data[0][i]);
    }
    for (i=0;i<index[1];i++) {
      fprintf(fp2,"\n%d",data[1][i]);
    }
    for (i=0;i<index[2];i++) {
      fprintf(fp3,"\n%d",data[2][i]);
    }
    for (i=0;i<index[3];i++) {
      fprintf(fp4,"\n%d",data[ ][i]);
    }
    for (i=0;i<index[4];i++) {
      fprintf(fp5,"\n%d",data[4][i]);
    }
    setcolor(EGA_DARKGREY);
    outtextxy(300,150,line_number);
}
fclose(fp1);
fclose(fp2);
fclose(fp3);
fclose(fp4);
fclose(fp5);
        setcolor(EGA_DARKGREY);
outtextxy(50,120,"Please wait for duration time of scanning.....");
setcolor(EGA_WHITE);
outtextxy(50,120,"Please wait for move back to begin.....");
        strcpy(command,"begin");    /* move head come back */
tx_data();
delay(2000);
        setcolor(EGA_DARKGREY);
outtextxy(50,120,"Please wait for move back to begin.....");
        setcolor(EGA_WHITE);
outtextxy(50,120,"Process is completed");
```

```
    outtextxy(50,180,"Press any key to continue");
            getch();
    titlewindow();
         }
         /** end scanning **/
         if ( func_key == 'h')  {
    helpwindow();
    titlewindow();
         }
         if ( func_key == 'q')   strcpy(command,"quit");
      } while (strcmp(command,"quit") != 0);
      confirm_exit();
      confirm = getch();
      if (confirm == 'y')   confirm = NULL;
      else   confirm = 'y';
    }
}

page_menu() {
    /* menu */
    setvisualpage(0);
    setactivepage(1);              /* set active page for write */
    setfillstyle(1,EGA_LIGHTGREY);
    bar(0,0,639,349);              /* external frame */
    setfillstyle(1,EGA_BLUE);
    bar(9,9,630,340);              /* internal frame */
    setfillstyle(1,EGA_LIGHTCYAN);
    bar(24,94,614,244);            /* external frame display */
    setfillstyle(1,EGA_DARKGREY);
    bar(29,99,609,239);            /* frame display */
    /********** menu description ***********/
    setcolor(EGA_LIGHTCYAN);
    settextstyle(0,0,2);
    outtextxy(X_MIN,40,"Metal Sheet Image Scanner System");
    settextstyle(2,0,7);
    /* left function */
    setcolor(EGA_WHITE);outtextxy(X_MIN+50,259,"A");
    setcolor(EGA_LIGHTRED); outtextxy(X_MIN+59,259,"lignment & calibrate");
    setcolor(EGA_WHITE);outtextxy(X_MIN+50,279,"S");
    setcolor(EGA_LIGHTRED);outtextxy(X_MIN+59,279,"can");
    setcolor(EGA_WHITE);outtextxy(X_MIN+50,299,"B");
    setcolor(EGA_LIGHTRED);outtextxy(X_MIN+59,299,"egin");
    setcolor(EGA_WHITE);outtextxy(X_MIN+50,319,"H");
    setcolor(EGA_LIGHTRED);outtextxy(X_MIN+59,319,"elp");
```

```
    /* right function */
    setcolor(EGA_WHITE);outtextxy(X_MIN+300,319,"Q");
    setcolor(EGA_LIGHTRED);outtextxy(X_MIN+309,319,"uit");
    /**************************************/
    setvisualpage(1);               /* set visual page for display */
}

confirm_run() {

    page_menu();
    setcolor(EGA_WHITE);
    outtextxy(50,130,"If you want to run this program.   You must be");
    outtextxy(50,150,"confirm by press 'Y' key or any key to force.");
}

confirm_exit() {

    page_menu();
    setcolor(EGA_WHITE);
    outtextxy(50,130,"If you want to exit from this program.   You must be");
    outtextxy(50,150,"confirm by press 'Y' key or any key to force.");
}

titlewindow() {

    /* text menu */
    page_menu();
    setcolor(EGA_WHITE);
    outtextxy(50,130,"If you want to run any function.   You can select");
    outtextxy(50,150,"by press highlight hot key.");
}
```

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

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

#define   DOT_PER_CM   200

FILE *fpc;
int  cam1[64],cam2[64],cam3[64],cam4[64],cam5[64];
int  index_cam1,index_cam2,index_cam3,index_cam4,index_cam5;
int  line_cam=1;
int  add_pos_cam1,add_pos_cam2,add_pos_cam3,add_pos_cam4,add_pos_cam5;
char config[] = "system.cfg";
int  i;
int  ch;
char value[30];
float Xc1,Xc2,Xc3,Xc4,Xc5;
int  Ofront1,Ofront2,Ofront3,Ofront4,Ofront5;
int  Oback1,Oback2,Oback3,Oback4,Oback5;
int  Yshift1,Yshift2,Yshift3,Yshift4,Yshift5;

read_config();
access_cam_file();
data_processing();

main() {

read_config();
access_cam_file();

}

read_config() {

 if ((fpc = fopen(config,"r")) != NULL) {
   i=0;
   while ((ch = getc(fpc)) != EOF)
     if (ch == '=') {
 fscanf(fpc,"%s\n",&value);
 if (i==0) Xc1 = atof(value);
 if (i==1) Xc2 = atof(value);
 if (i==2) Xc3 = atof(value);
 if (i==3) Xc4 = atof(value);
 if (i==4) Xc5 = atof(value);
 if (i==5) Yshift1 = atoi(value);
 if (i==6) Yshift2 = atoi(value);
```

```
  if (i==7) Yshift3 = atoi(value);
  if (i==8) Yshift4 = atoi(value);
  if (i==9) Yshift5 = atoi(value);
  if (i==10) Ofront1 = atoi(value);
  if (i==11) Ofront2 = atoi(value);
  if (i==12) Ofront3 = atoi(value);
  if (i==13) Ofront4 = atoi(value);
  if (i==14) Ofront5 = atoi(value);
  i++;
     }
   fclose(fpc);
  }
  else
    printf("error in opening file.");
  /* end of line scan */
  Oback1 = Ofront1 + (20 * DOT_PER_CM);
  Oback2 = Ofront2 + (20 * DOT_PER_CM);
  Oback3 = Ofront3 + (20 * DOT_PER_CM);
  Oback4 = Ofront4 + (20 * DOT_PER_CM);
  Oback5 = Ofront5 + (20 * DOT_PER_CM);
  add_pos_cam1 = 0;
  add_pos_cam2 = 4000;
  add_pos_cam3 = 8000;
  add_pos_cam4 = 12000;
  add_pos_cam5 = 16000;
  /* reference for connect data */
}

access_cam_file() {

FILE *fp1,*fp2,*fp3,*fp4,*fp5,*fp_out;

   fp1 = fopen("cam1.scn","r");
   fp2 = fopen("cam2.scn","r");
   fp3 = fopen("cam3.scn","r");
   fp4 = fopen("cam4.scn","r");
   fp5 = fopen("cam5.scn","r");
   fp_out = fopen("cam.tot","w");
   fscanf(fp1,"%s\n",&value);   /* return for title name */
   fscanf(fp2,"%s\n",&value);
   fscanf(fp3,"%s\n",&value);
   fscanf(fp4,"%s\n",&value);
   fscanf(fp5,"%s\n",&value);
   fprintf (fp_out,"FILE_cam.tot");   /* title File */
```

```
fscanf(fp1,"%s\n",&value);    /* feed drain */
fscanf(fp2,"%s\n",&value);    /* feed drain */
fscanf(fp3,"%s\n",&value);    /* feed drain */
fscanf(fp4,"%s\n",&value);    /* feed drain */
fscanf(fp5,"%s\n",&value);    /* feed drain */
/*****************************************************************/
/* use Yshift for feed shift to drain */
/* read file cam1 */
for (i=0;i < Yshift1;i++) {
  fscanf(fp1,"%s\n",&value);    /* jump line of cam1 */
  fscanf(fp1,"%s\n",&value);
  while ( (value[0] != 'S') && (!feof(fp1))) {
    fscanf(fp1,"%s\n",&value);
  }
}
/* read file cam2 */
for (i=0;i < Yshift2;i++) {
  fscanf(fp2,"%s\n",&value);    /* jump line of cam2 */
  fscanf(fp2,"%s\n",&value);
  while ( (value[0] != 'S') && (!feof(fp2))) {
    fscanf(fp2,"%s\n",&value);
  }
}
/* read file cam3 */
for (i=0;i < Yshift3;i++) {
  fscanf(fp3,"%s\n",&value);    /* jump line of cam3 */
  fscanf(fp3,"%s\n",&value);
  while ( (value[0] != 'S') && (!feof(fp3))) {
    fscanf(fp3,"%s\n",&value);
  }
}
/* read file cam4 */
for (i=0;i < Yshift4;i++) {
  fscanf(fp4,"%s\n",&value);    /* jump line of cam4 */
  fscanf(fp4,"%s\n",&value);
  while ( (value[0] != 'S') && (!feof(fp4))) {
    fscanf(fp4,"%s\n",&value);
  }
}
/* read file cam5 */
for (i=0;i < Yshift5;i++) {
  fscanf(fp5,"%s\n",&value);    /* jump line of cam5 */
  fscanf(fp5,"%s\n",&value);
  while ( (value[0] != 'S') && (!feof(fp5))) {
```

```
      fscanf(fp5,"%s\n",&value);
    }
  }
/*****************************************************************/
  while ( (!feof(fp1)) && (!feof(fp2)) && (!feof(fp3)) && (!feof(fp4)) &&
  (!feof(fp5)) ) {   /* check end of file of five cam */
  /* read file cam1 */       /* final line may be loss because it have */
    i=0;                     /* line non transition data */
    fscanf(fp1,"%s\n",&value);   /* jump line of cam1 */
    fscanf(fp1,"%s\n",&value);
    while ( (value[0] != 'S') && (!feof(fp1)) ) {
      cam1[i] = atoi(value);
      i++;
      fscanf(fp1,"%s\n",&value);
    }
    index_cam1 = i;
  /* read file cam2 */
    i=0;
    fscanf(fp2,"%s\n",&value);   /* jump line of cam2 */
    fscanf(fp2,"%s\n",&value);
    while ( (value[0] != 'S') && (!feof(fp2)) ) {
      cam2[i] = atoi(value);
      i++;
      fscanf(fp2,"%s\n",&value);
    }
    index_cam2 = i;
  /* read file cam3 */
    i=0;
    fscanf(fp3,"%s\n",&value);   /* jump line of cam3 */
    fscanf(fp3,"%s\n",&value);
    while ( (value[0] != 'S') && (!feof(fp3)) ) {
      cam3[i] = atoi(value);
      i++;
      fscanf(fp3,"%s\n",&value);
    }
    index_cam3 = i;
  /* read file cam4 */
    i=0;
    fscanf(fp4,"%s\n",&value);   /* jump line of cam4 */
    fscanf(fp4,"%s\n",&value);
    while ( (value[0] != 'S') && (!feof(fp4)) ) {
      cam4[i] = atoi(value);
      i++;
      fscanf(fp4,"%s\n",&value);
```

```
  }
  index_cam4 = i;
/* read file cam5 */
  i=0;
  fscanf(fp5,"%s\n",&value);   /* jump line of cam5 */
  fscanf(fp5,"%s\n",&value);
  while ( (value[0] != 'S') && (!feof(fp5))) {
    cam5[i] = atoi(value);
    i++;
    fscanf(fp5,"%s\n",&value);
  }
  index_cam5 = i;
data_processing();
/*********************************************************/
/* write line number */
fprintf (fp_out,"\nS%d",line_cam);
line_cam++;
/* write cam 1 to output file cam.tot */
for (i=0;i<index_cam1;i++) {
  fprintf(fp_out,"\n%d",cam1[i]);
}
/* write cam 2 to output file cam.tot */
for (i=0;i<index_cam2;i++) {
  fprintf(fp_out,"\n%d",cam2[i]);
}
/* write cam 3 to output file cam.tot */
for (i=0;i<index_cam3;i++) {
  fprintf(fp_out,"\n%d",cam3[i]);
}
/* write cam 4 to output file cam.tot */
for (i=0;i<index_cam4;i++) {
  fprintf(fp_out,"\n%d",cam4[i]);
}
/* write cam 5 to output file cam.tot */
for (i=0;i<index_cam5;i++) {
  fprintf(fp_out,"\n%d",cam5[i]);
}
/*********************************************************/
}
fclose(fp1);
fclose(fp2);
fclose(fp3);
fclose(fp4);
fclose(fp5);
```

```
    fclose(fp_out);
}

data_processing() {

int count;
/********/
/* cam1 */
/********/
  for (i=0,count=0;i<index_cam1;i++) {
    cam1[i] = (float)cam1[i] * Xc1;
    if ( ( cam1[i] < 0back1 ) && (cam1[i] >= 0front1) ) {
      cam1[count] = cam1[i] - 0front1 + add_pos_cam1;
      count++;
    }
  }
  index_cam1 = count;
/********/
/* cam2 */
/********/
  for (i=0,count=0;i<index_cam2;i++) {
    cam2[i] = (float)cam2[i] * Xc2;
    if ( ( cam2[i] < 0back2 ) && (cam2[i] >= 0front2) ) {
      cam2[count] = cam2[i] - 0front2 + add_pos_cam2;
      count++;
    }
  }
  index_cam2 = count;
/********/
/* cam3 */
/********/
  for (i=0,count=0;i<index_cam3;i++) {
    cam3[i] = (float)cam3[i] * Xc3;
    if ( ( cam3[i] < 0back3 ) && (cam3[i] >= 0front3) ) {
      cam3[count] = cam3[i] - 0front3 + add_pos_cam3;
      count++;
    }
  }
  index_cam3 = count;
/********/
/* cam4 */
/********/
  for (i=0,count=0;i<index_cam4;i++) {
    cam4[i] = (float)cam4[i] * Xc4;
```

```
    if ( ( cam4[i] < Oback4 ) && (cam4[i] >= Ofront4) ) {
      cam4[count] = cam4[i] - Ofront4 + add_pos_cam4;
      count++;
    }
  }
  index_cam4 = count;
/********/
/* cam5 */
/********/
  for (i=0,count=0;i<index_cam5;i++) {
    cam5[i] = (float)cam5[i] * Xc5;
    if ( ( cam5[i] < Oback5 ) && (cam5[i] >= Ofront5) ) {
      cam5[count] = cam5[i] - Ofront5 + add_pos_cam5;
      count++;
    }
  }
  index_cam5 = count;
}
```

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

```
#include <z80io.h>
#include <string.h>

#define  DATA_PORT          0x30      /* port d */
#define  DATA_DIR_PORT      0x34      /* dir port d */
#define  DIR_INPUT          0x00
#define  DIR_OUTPUT         0xFF
#define  CONTROL_PORT       0x40      /* port E */
#define  CONTROL_DIR_PORT   0x44      /* dir port E */
#define  BUS_PORT           0x50      /* port A */
#define  BUS_DIR_PORT       0X54      /* dir port A */
#define  STATUS_PORT        0x51      /* port B */
#define  STATUS_DIR_PORT    0x55      /* dir port B */
#define  SPECIAL_DIR_PORT   0x56      /* dir port C */
#define  SPECIAL_PORT       0x52      /* port C */
#define  END_OF_COMMAND     0x00
#define  END_OF_PARAMETER   0x00
#define  END_OF_MESSAGE     0xFF
#define  END_OF_STATUS      0xFF
#define  NULL               0x00
#define  N_ST_SG_RX         0x00
#define  ST_SG_RX           0x02
#define  N_ST_SG_TX         0x04
#define  ST_SG_TX           0x06
#define  MASK_BIT_0         0x01
#define  MASK_BIT_1         0x02
#define  MASK_BIT_4         0x10
#define  MASK_BIT_45        0x30
#define  ZERO               0
#define  SPACE              '/'
#define  BREAK_OK           0x00
#define  BREAK_NOK          0x01
         /* CCD */
#define  SET_VOLT           0x80
#define  START_BRD_SYS      0x02  /* pulse */
#define  RESET_BRD_SYS      0x04  /* pulse */
#define  CLEAR              0x00
#define  NOT_CLEAR          0x08
#define  PULSE_SEL_CAM      0x08
#define  BYTE_H             0x04
#define  BYTE_L             0x00
#define  INC_READ           0x40
#define  EMPTY_DATA         0x00
```

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

```
#define  NORMAL_DATA        0x30
#define  FAIL_DATA1         0x66  /* fail code (error)*/
#define  FAIL_DATA2         0x67  /* fail code (over)*/
         /* STEPPER MOTOR */
#define  DRV_FORWARD        0x30  /* bit 6 remote if hi for energize */
#define  NOT_DRV_FORWARD    0x20
#define  DRV_REVERSE        0x10  /* bit 5 dir of rotation */
#define  NOT_DRV_REVERSE    0x00
#define  MOTOR_HOLD         0x00  /* bit 4 start motor */
         /* CTC */
#define  CHANNEL_0          0x10   /* CTC Address Port */
#define  CHANNEL_1          0x11
#define  CHANNEL_2          0x12
#define  CON_WRD_0          0x4D   /* 0100 1101 */
#define  TIM_CNT_0          20
#define  CON_WRD_1          0xCF   /* 1100 1111 */
#define  TIM_CNT_1          250
#define  CON_WRD_2          0x5F   /* 0101 1111 */
#define  TIM_CNT_2          36
         /* COMMAND */
#define  COMMAND1           "scan"
#define  COMMAND2           "threshold"
#define  COMMAND3           "ali&cal"
#define  COMMAND4           "begin"

void mainfunc();
void init_io();
void RX_DATA();
void TX_DATA();
void select();
void read_data();
void threshold();
void endscan();

char  command[14],parameter[4],message[130][5],status[2][5],process;
char  read_out,common_special,clear_fifo,act_motor;

/* <<< end of critical region >>> */
main() {
#entry
  org   0h
  di
  ld    sp,9ff0h
  jp    mainfunc
```

```
#endasm

#asm
  ;**** interrup service ****
  org   38h
  di
  push  af
  push  bc
  push  de
  push  hl
/******************/
/* begin counter */
/******************/
  /* CTC CHANNEL 0 */
  ld    a,040h
  out   (10h),a           /* 0100 1101 */
  ld    a,20
  out   (10h),a
  /* CTC CHANNEL 1 */
  ld    a,0CFh
  out   (11h),a           /* 1100 1111 */
  ld    a,250
  out   (11h),a
  /* CTC CHANNEL 2 */
  ld    a,05Fh
  out   (12h),a           /* 0101 1111 */
  ld    a,36
  out   (12h),a
/******************/
      ld    a,(read_out)      /* var read_out in reg e */
      ld    e,a
      ld    a,(act_motor)
      or    e     /* store process = read_out | active_motor value in reg h */
      ld    h,a
/*********************************************************/
      ld    d,08              /* NOT_CLEAR */
/**/  ld    a,04h             /* RESET_BRD_SYS      04 */
      or    d     /* common_special = process | RESET_BRD_SYS | NOT_CLEAR */
      or    h
      out   (52h),a           /* reset ccd system */

/* clear_fifo = (read_out ^ BREAK_NOK) << 3 */
/**/  ld    a,01              /* BREAK_NOK */
      xor   e                 /* read_out ^ BREAK_NOK */
```

```
        rlca                    /* << 3 */
        rlca
        rlca
/**/    or    'h  /* common_special = process | clear_fifo */
        out   (52h),a           /* out for clear_fifo */

/**/    ld    a,02              /* START_BRD_SYS      02 */
        or    d     /* common_special = process | START_BRD_SYS | NOT_CLEAR */
        or    h
        out   (52h),a           /* start for read ccd */

/**/    ld    a,d               /* common_special = process | NOT_CLEAR */
        or    h
        out   (52h),a
        ld    a,e
        ld    (process),a       /* confirm process for checking */
  pop   hl
  pop   de
  pop   bc
  pop   af
  ei
  reti
#endasm
}
/* <<< end of critical region >>> */

mainfunc() {
  char  count;

  read_out = BREAK_OK;
  act_motor = MOTOR_HOLD;
  intrmode_1();                 /* set interrupt mode 0 */
  for (count=0;count<10;count++);
  init_io();
  for (count=0;count<10;count++);
  endscan();                     /* begin counter */
  while (1) {
    RX_DATA();
    select();
    act_motor = MOTOR_HOLD;
  }
}

init_io() {
```

```
   /**************/
   /* init port */
   /**************/
   outport(CONTROL_DIR_PORT,0x06); /* E2(dir,O/P),E1(strobe,O/P),E0(busy,I/P)*/
   outport(STATUS_DIR_PORT,0xCC); /* ooiiooii  PORT B */
     /* SET_VOLT,INC_READ,/FULL,/EMPTY,SEL_CAM,BYTE_L_H,REF,ENCODER */
   outport(SPECIAL_DIR_PORT,DIR_OUTPUT);  /* oooooooo */
     /* port c X,REMOTE,MOTOR_DIR,START_STEP,/CLR_FIFO,RESET,START,/BREAK */
}

RX_DATA() {
   char  i,temp;

   outport(DATA_DIR_PORT,DIR_INPUT);
   i=0;
   do {
      outport(CONTROL_PORT,ST_SG_RX);                    /* give HI */
      while(((char)inport(CONTROL_PORT) & MASK_BIT_0) == ZERO ); /* HI pass */
      command[i] = (char)inport(DATA_PORT);                  /* read data */
      outport(CONTROL_PORT,N_ST_SG_RX);                  /* give LO */
      while(((char)inport(CONTROL_PORT) & MASK_BIT_0) > ZERO );  /* LO pass */
      i++;
      temp = command[i-1];
      if (temp == SPACE) temp = END_OF_COMMAND;
   } while (temp != END_OF_COMMAND);
   /* read parameter part */
   if (command[i-1] == SPACE) {
     command[i-1] = END_OF_COMMAND;
     i=0;
     do {
      outport(CONTROL_PORT,ST_SG_RX);                   /* give HI */
      while(((char)inport(CONTROL_PORT) & MASK_BIT_0) == ZERO );/* HI pass */
      parameter[i] = (char)inport(DATA_PORT);                /* read data */
      outport(CONTROL_PORT,N_ST_SG_RX);                 /* give LO */
      while(((char)inport(CONTROL_PORT) & MASK_BIT_0) > ZERO ); /* LO pass */
      i++;
     } while(parameter[i-1] != END_OF_PARAMETER);
   }
}

TX_DATA() {
  char  i,toggle,cam,temp;
```

```
  outport(DATA_DIR_PORT,DIR_OUTPUT);
 for (cam=0;cam<5;cam++) {
  i = 0; toggle = ZERO;
     /* TX data */
    do {
      if (toggle == ZERO) toggle = 0xFF;
      else toggle = 0;
      while (((char)inport(CONTROL_PORT) & MASK_BIT_0 ) == ZERO );  /* HI pass */
      outport(DATA_PORT,message[i][cam]);        /* write data to printer port */
      outport(CONTROL_PORT,ST_SG_TX);                              /* give HI */
      while (((char)inport(CONTROL_PORT) & MASK_BIT_0 ) > ZERO );  /* LO pass */
      outport(CONTROL_PORT,N_ST_SG_TX);                            /* give LO */
      i++;
      if (toggle == ZERO) temp = 0;
      else temp = message[i-1][cam];
    } while (temp != END_OF_MESSAGE);
    /* TX status */
    i=0;
    do {
      while (((char)inport(CONTROL_PORT) & MASK_BIT_0 ) == ZERO );  /* HI pass */
      outport(DATA_PORT,status[i][cam]);        /* write data to printer port */
      outport(CONTROL_PORT,ST_SG_TX);                              /* give HI */
      while (((char)inport(CONTROL_PORT) & MASK_BIT_0 ) > ZERO );  /* LO pass */
      outport(CONTROL_PORT,N_ST_SG_TX);                            /* give LO */
      i++;
    } while (status[i-1][cam] != END_OF_STATUS);
}
}

select() {
  int  j,convert;

  if (strcmp(command,COMMAND1) == 0) {
     parameter[0] = (parameter[0]-'0')*100 + (parameter[1]-'0')*10 + (parameter[2]-'0');
     convert = (int)parameter[0] * 50;
     for (j=0;j < convert;j++) {
        read_out = BREAK_NOK;     /* UNBREAK */
        while (process == BREAK_OK);
        read_out = BREAK_OK;      /* BREAK FOR READING */
        while (process == BREAK_NOK);
        read_data();
        act_motor = DRV_FORWARD;  /* move head in transmitt during */
        TX_DATA();                /* transmitt data to computer */
        while (((char)inport(STATUS_PORT) & MASK_BIT_0) != ZERO ); /*check end drv*/
```

```
        act_motor = NOT_DRV_FORWARD;
      }
  }
  if (strcmp(command,COMMAND2) == 0) {
        threshold();
  }
  if (strcmp(command,COMMAND3) == 0) {
        read_out = BREAK_NOK;
        while (process == BREAK_OK);
        read_out = BREAK_OK;
        while (process == BREAK_NOK);
        read_data();
        TX_DATA();
  }
  if (strcmp(command,COMMAND4) == 0) {
        while (((char)inport(STATUS_PORT) & MASK_BIT_1 ) != ZERO) { /* check refer */
          act_motor = DRV_REVERSE;
          while (((char)inport(STATUS_PORT) & MASK_BIT_0) != ZERO );/*check end drv*/
          act_motor = NOT_DRV_REVERSE;
        }
  }
}

read_data() {    /* tranfer data to Microcontroller */
  char  i,cam,over;

  outport(BUS_DIR_PORT,DIR_INPUT);
  /* read status */
 for (cam=0;cam<5;cam++) {
  outport(STATUS_PORT,NULL);
  outport(STATUS_PORT,PULSE_SEL_CAM);
  outport(STATUS_PORT,NULL);
  i=0;
  status[i][cam] = (char)inport(STATUS_PORT);
  i++;
  status[i][cam] = END_OF_STATUS;
  i=0;
  if( (status[0][cam] & MASK_BIT_45) == NORMAL_DATA ) {
     do {
       outport(STATUS_PORT,BYTE_H);        /* read byte high */
       message[i][cam] = (char)inport(BUS_PORT);
       i++;
       outport(STATUS_PORT,NULL);
       outport(STATUS_PORT,BYTE_L);        /* read byte low */
```

```
        message[i][cam] = (char)inport(BUS_PORT);
        i++;
        outport(STATUS_PORT,NULL);
        outport(STATUS_PORT,INC_READ);      /* inc for next read */
        outport(STATUS_PORT,NULL);
        over = ((char)inport(STATUS_PORT) & MASK_BIT_4);
        if (message[i-2][cam] == 0xFF) {  /* check fail data */
           over = 0x00;
           status[0][cam] = FAIL_DATA1;
           i=0;
        }
        if (i>125) {                    /* check over */
           over = 0x00;
           status[0][cam] = FAIL_DATA2;
           i=0;
        }
      } while (over != EMPTY_DATA );
      message[i][cam] = END_OF_MESSAGE;
      i++;
   }
   else message[0][cam] = END_OF_MESSAGE;
  }
 }

 threshold() {

   outport(BUS_DIR_PORT,DIR_OUTPUT);
   parameter[0] = (parameter[0]-'0')*100 + (parameter[1]-'0')*10 + (parameter[2]-'0');
   outport(BUS_PORT,parameter[0]);    /* value of voltage */
   outport(STATUS_PORT,NULL);
   outport(STATUS_PORT,SET_VOLT);      /* set voltage */
   outport(STATUS_PORT,NULL);
 }

 endscan() {

 #asm
   push  af
   push  bc
   push  de
   push  hl
 /******************/
 /* begin counter */
 /******************/
```

```
   /* CTC CHANNEL 0 */
   ld    a,04Dh
   out   (10h),a          /* 0100 1101 */
   ld    a,20
   out   (10h),a
   /* CTC CHANNEL 1 */
   ld    a,0CFh
   out   (11h),a          /* 1100 1111 */
   ld    a,250
   out   (11h),a
   /* CTC CHANNEL 2 */
   ld    a,05Fh
   out   (12h),a          /* 0101 1111 */
   ld    a,36
   out   (12h),a
 /******************/
      ld    a,(read_out)      /* var read_out in reg e */
      ld    e,a
      ld    a,(act_motor)
      or    e     /* store process = read_out | active_motor value in reg h */
      ld    h,a
 /*********************************************************/
      ld    d,08              /* NOT_CLEAR */
/**/  ld    a,04h             /* RESET_BRD_SYS      04 */
      or    d     /* common_special = process | RESET_BRD_SYS | NOT_CLEAR */
      or    h
      out   (52h),a           /* reset ccd system */

/* clear_fifo = (read_out ^ BREAK_NOK) << 3 */
/**/  ld    a,01              /* BREAK_NOK */
      xor   e                 /* read_out ^ BREAK_NOK */
      rlca                    /* << 3 */
      rlca
      rlca
/**/  or    h  /* common_special = process | clear_fifo */
      out   (52h),a           /* out for clear_fifo */

/**/  ld    a,02              /* START_BRD_SYS      02 */
      or    d     /* common_special = process | START_BRD_SYS | NOT_CLEAR */
      or    h
      out   (52h),a           /* start for read ccd */

/**/  ld    a,d               /* common_special = process | NOT_CLEAR */
      or    h
```

```
        out   (52h),a
        ld    a,e
        ld    (process),a       /* confirm process for checking */
    pop   hl
    pop   de
    pop   bc
    pop   af
    ei
#endasm

}
```

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ประวัติผู้เขียน

นาย ปรเมษฐ์ จินุพันธุ์ เกิดวันที่ 1 กรกฎาคม พ.ศ.2510 ที่อำเภอเมืองจังหวัดสงขลา สำเร็จการศึกษาปริญาตรีวิศวกรรมศาสตรบัณฑิต สาขาอิเล็กทรอนิกส์-สื่อสาร ภาควิชา วิศวกรรมไฟฟ้า คณะวิศวกรรมเทคโนโลยี สถาบันเทคโนโลยีราชมงคล วิทยาเขต เทเวศร์ ในปีการศึกษา 2532 และเข้าศึกษาต่อในหลักสูตรวิศวกรรมศาสตรมหาบัณฑิต ที่ จุฬาลงกรณ์ มหาวิทยาลัย เมื่อ พ.ศ.2534

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย