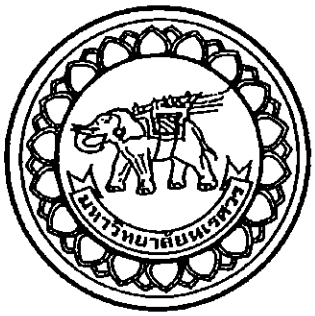


เครื่องคัดแยกผลิตผลทางการเกษตร
ด้วยกระบวนการวิเคราะห์ภาพ

Automatic separator by

Image Processing

ห้องสมุดคณะวิศวกรรมศาสตร์
วันที่ ... 12 6 เม.ย. 2544 ...
เลขทะเบียน ... ดศ 4400199 ...
เลขเรียกหนังสือ ... S ...
มหาวิทยาลัยนเรศวร
675
ส4750
2543

I 5091172
ปร.
ส475ค
2543
ค.2

นางสาวจินดารักษ์ กลีบจินดา รหัส 40360232

นายสถิตพงษ์ รสขวา รหัส 40360521

โครงงานนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

ปีการศึกษา 2543

ใบรับรองโครงงานวิจัย

หัวข้อโครงงาน	:	เครื่องคัดแยกผลิตผลทางการเกษตรด้วยกระบวนการวิเคราะห์ภาพ
Performance	:	Automatic Separator by Image Processing
ผู้ดำเนินโครงงาน	:	นางสาวจินดารักษ์ กลีบจินดา รหัส 40360232
		นายสถิตพงษ์ รสขวา รหัส 40360521
อาจารย์ที่ปรึกษา	:	อ. ปัญญา เหล่าอนันต์ธนา
อาจารย์ที่ปรึกษาร่วม	:	อ. มุฑิตา สงฆ์จันทร์
สาขา	:	วิศวกรรมคอมพิวเตอร์
ภาควิชา	:	วิศวกรรมไฟฟ้าและคอมพิวเตอร์

.....

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร อนุมัติให้โครงงานฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร วิศวกรรมศาสตร์บัณฑิต สาขาวิศวกรรมคอมพิวเตอร์

คณะกรรมการสอบโครงงานวิจัย

.....ประธานกรรมการ
(อาจารย์ปัญญา เหล่าอนันต์ธนา)

.....กรรมการ
(อาจารย์พนัส นัถฤทธิ์)

.....กรรมการ
(อาจารย์มุฑิตา สงฆ์จันทร์)

หัวข้อโครงงาน	: เครื่องคัดแยกผลิตผลทางการเกษตรด้วยกระบวนการวิเคราะห์
ผู้ดำเนินโครงงาน	: นางสาวจินดารักษ์ กลีบจินดา รหัส 40360232
	นายสถิตพงษ์ รสขวา รหัส 40360521
อาจารย์ที่ปรึกษา	: อ. ปัญญา เหล่าอนันต์ธนา
อาจารย์ที่ปรึกษาร่วม	: อ. มุฑิตา สงฆ์จันทร์
สาขา	: วิศวกรรมคอมพิวเตอร์
ภาควิชา	: วิศวกรรมไฟฟ้าและคอมพิวเตอร์
ปีการศึกษา	: 2543

.....

บทคัดย่อ

โครงงานนี้จัดทำเกี่ยวกับการศึกษาการคัดแยกวัตถุด้วยกระบวนการวิเคราะห์ภาพ, การออกแบบและสร้างชุดควบคุมมอเตอร์, การออกแบบและสร้างชุดตรวจสอบวัตถุ และนำผลการศึกษาที่ได้ทั้งหมดมาออกแบบและสร้างเครื่องคัดแยกผลิตผลทางการเกษตรซึ่งในโครงงานจะใช้ผลมะม่วงเป็นตัวสาธิต โดยเครื่องคัดแยกจะทำการคัดแยกผลมะม่วงออกเป็น 3 ขนาด ด้วยเกณฑ์ที่ใช้การวิเคราะห์ภาพผลมะม่วงเป็นตัวบอกว่ามะม่วงแต่ละผลนั้นอยู่ในเกณฑ์ขนาดใดเพื่อให้เครื่องคัดแยกผลมะม่วงนั้นไปตามทางลำเลียงลงสู่ตระกร้าที่เหมาะสมกับขนาดของผลมะม่วง

Project Title	:	Automatic Separator by Image Processing
Name	:	Ms.Jindaruk Gleebjinda ID. 40360232
		Mr.Satitpong Roskuar ID. 40360521
Project Advisor	:	Mr.Panya Loawanuntana
Co- Project Advisor	:	Ms.Mutita Songjun
Field of Study	:	Computer Engineering
Department	:	Electrical and Computer Engineering
Academic Year	:	2000

Abstract

The content of this project is about studying separate object with Image Process, design and build the drivenmotor, design and build the sensor. Then design and built the automatic Separator by Image Process with the result from studying. This project use mango in presenting that the Separator separates mangoes for three size by Image Process.

กิตติกรรมประกาศ

โครงงานนี้ดำเนินการงานสำเร็จมาได้เนื่องจากความอนุเคราะห์ของทุกท่านเหล่านี้ อาจารย์ปัญญา เหล่าอนันต์ธนา อาจารย์ที่ปรึกษาโครงงาน, อาจารย์ประเทือง โมรารายา อาจารย์ที่ให้คำชี้แนะและความช่วยเหลือด้านการเชื่อมตัวโครงเหล็กของเครื่องคัดแยกผลมะม่วง, และเพื่อนๆชมรมโรบอทที่เอื้อเฟื้อความช่วยเหลือด้านอุปกรณ์และสถานที่ในการทำงาน ขอขอบคุณทุกท่านที่มีส่วนช่วยให้โครงงานชิ้นนี้สำเร็จลุล่วงได้ด้วยดี

น.ส.จินดารักษ์ กลีบจินดา

นายสถิตพงษ์ รสขวา

ผู้จัดทำโครงงาน

สารบัญ

	หน้า
ใบรับรองโครงงานวิจัย	ก
บทคัดย่อ	ข
ABSTRACT	ค
กิตติกรรมประกาศ	ง
สารบัญ	จ
สารบัญตาราง	ช
สารบัญรูปภาพ	ซ
บทที่ 1 บทนำ	1
1.1 ที่มาและความสำคัญของโครงงาน	1
1.2 วัตถุประสงค์ของโครงงาน	1
1.3 ขอบข่ายของโครงงาน	2
1.4 ขั้นตอนการดำเนินงาน	2
1.5 ผลที่คาดว่าจะได้รับ	3
1.6 งบประมาณที่ใช้	3
บทที่ 2 หลักการและทฤษฎี	4
2.1 หลักการทางเครื่องกล	4
2.2 ระบบควบคุมมอเตอร์	4
2.3 ระบบ sensor	5
2.4 ชุดควบคุมการกลับสู่ตำแหน่งของชุดผลัก	6
2.5 การทำงานของชุดคัดแยก	7
2.6 การสื่อสารข้อมูลอนุกรม	11
2.7 การอินเตอร์รัปต์ของการสื่อสารการอนุกรม	12
2.8 กระบวนการรับและส่งข้อมูลอนุกรม	12
2.9 การทำงานของการสื่อสารทางพอร์ตอนุกรม	13
2.10 การติดตั้งการสื่อสารทางพอร์ตอนุกรมด้านฮาร์ดแวร์	14
2.11 Microcontroller 8031	15
2.12 รูปแบบข้อมูลภาพโดยทั่วไป	20

สารบัญ(ต่อ)

	หน้า
2.13 Algorithm การวิเคราะห์ภาพ	21
บทที่ 3 วิธีดำเนินงาน	22
3.1 การดำเนินโครงงาน	22
3.2 การออกแบบ	23
3.3 การจัดสร้างชิ้นงาน	29
3.4 ขั้นตอนการทดลอง	31
บทที่ 4 ผลการทดลอง และผลการวิเคราะห์	32
4.1 ผลการทดลอง	32
4.2 วิเคราะห์ผลการทดลอง	33
บทที่ 5 บทสรุป	34
5.1 สรุปผลการทดลอง	34
5.2 ข้อเสนอแนะ	34
5.3 ปัญหาที่เกิดขึ้นในระหว่างการจัดทำโครงงานและการแก้ไข	35
บรรณานุกรม	36
ภาคผนวก	37
ส่วนของโปรแกรมคอนโทรเลอร์	38
ส่วนของโปรแกรมการวิเคราะห์ภาพ	52
Datasheet เกี่ยวกับเรื่อง Timer 555	66
Datasheet เกี่ยวกับเรื่อง TSOP18..	78
ประวัติผู้จัดทำ	84

สารบัญตาราง

	หน้า
ตารางที่ 1.1 ตารางกิจกรรมการดำเนินงาน	2
ตารางที่2.1 ตารางแสดงคอนเน็คเตอร์ของพอร์ต RS-232C แบบ 9 ขา	14
ตารางที่4.1 ตารางแสดงผลการทดลองของมะม่วงพันธุ์แรก	32
ตารางที่4.2 ตารางแสดงผลการทดลองของมะม่วงพันธุ์ที่สอง	32

มหาวิทยาลัยนเรศวร

สารบัญรูปภาพ

	หน้า
รูปที่2.1 ภาพแสดงวงจรควบคุมมอเตอร์	4
รูปที่2.2 ภาพแสดงวงจรภาคส่งของ sensor	5
รูปที่2.3 ภาพแสดงวงจรภาครับของ sensor	5
รูปที่2.4 ภาพแสดงวงจรชุดควบคุมความถี่ของ sensor	6
รูปที่2.5 ภาพแสดงผังกระบวนการทำงานของโปรแกรมวิเคราะห์ภาพ	7
รูปที่2.6 ภาพแสดงผังการทำงานของโปรแกรม Controller	8
รูปที่2.7 ภาพแสดงตำแน่ง ณ.จุดต่างๆ ของเครื่องคัดแยก	10
รูปที่2.8 ภาพแสดงการต่อเข้ากับคอมพิวเตอร์โดยต่อเข้าทาง PRINTER PORT โดยใช้สาย DB25 PIN	19
รูปที่3.1 ภาพแสดงแบบโครงสร้างของเครื่องคัดแยก	23
รูปที่3.2 ภาพแสดงวงจรควบคุมมอเตอร์ของชุดสายพานและชุดผลัก	25
รูปที่3.3 ภาพแสดงวงจรชุด sensor และ วงจรของ sensor limit switch	27
รูปที่3.4 ภาพแสดงส่วนของการแสดงผลและการติดต่อกับผู้ใช้	28

บทที่1

บทนำ

การคัดแยกวัตถุหลากหลายขนาดและรูปแบบ รวมทั้งการคัดแยกผลิตผลทางการเกษตรออกหลายขนาด ปัจจุบันนั้นใช้เครื่องจักรหรือแรงงานมนุษย์ แต่ในการแยกผลไม้ที่สุกหรือดิบจะใช้แรงงานเครื่องจักรไม่ได้ จึงใช้ได้เพียงแรงงานมนุษย์ ด้วยเหตุนี้จึงทำให้อาจมีปัญหาเกิดขึ้นตามมาได้ เช่นปัญหาต่อไปนี้ ความไม่แน่นอนในการคัดแยก, มาตรฐานในการคัดแยกไม่มี, และความสามารถจำกัดในการทำงานเนื่องจากมนุษย์เรามีความเหน็ดเหนื่อยเมื่อยล้าได้ ดังนั้นหากเราสามารถทำให้เครื่องจักรทำงานนี้แทนมนุษย์ได้ก็จะทำให้งานที่ได้มีความแน่นอนตามมาตรฐานที่ถูกตั้งไว้โดยเราเป็นผู้กำหนดมาตรฐานที่เราต้องการและสั่งงานให้เครื่องจักรนำไปทำงานแทนการใช้แรงงานมนุษย์ได้

1.1 ที่มาและความสำคัญของโครงงาน

โครงงานนี้ทำการคัดแยกวัตถุโดยใช้ขนาดเป็นเกณฑ์ ขนาดจะถูกแบ่งด้วยการประมวลผลที่อาศัยหลักการ Image Process จากที่กล่าวมาแล้วในบทนำนั้นจะพบว่าในการคัดแยกผลผลิตทางการเกษตรนั้นยังต้องการการพัฒนาอยู่ โดยในโครงงานนี้จะสาธิตการคัดแยกผลมะม่วง โดยใช้หลักการ Image Process ไปพัฒนาเป็นโปรแกรม เพื่อคัดแยกผลมะม่วงขนาดต่างๆ ซึ่งผู้จัดทำโครงงานหวังว่าผลจากโครงงานนี้จะเป็นก้าวไปสู่การพัฒนาด้านแรงงานเครื่องจักรในการคัดแยกผลผลิตทางการเกษตรให้มีประสิทธิภาพมากยิ่งขึ้น เพื่อประหยัดแรงงานของมนุษย์ และเวลาอีกทั้งยังได้ความแม่นยำแน่นอนตามเกณฑ์มาตรฐานที่กำหนด

1.2 วัตถุประสงค์

1. เพื่อศึกษาถึงระบบการวิเคราะห์ภาพ (Image Processing)
2. สามารถนำหลักการวิเคราะห์ภาพไปประยุกต์ใช้
3. สามารถเห็นถึงการนำไปใช้งานได้จริง
4. เพื่อเป็นแนวทางในการนำไปพัฒนาและประยุกต์ใช้เพื่อให้เกิดประโยชน์

1.3 ขอบข่ายของงาน

- วิเคราะห์ภาพโดยสามารถที่จะหาขอบของภาพที่ต้องการ เพื่อหาขนาดในระบบ 2 มิติ
- วิเคราะห์ภาพโดยหาปริมาณสีที่แตกต่างกันภายในภาพ เพื่อบอกถึงความสุกของผลิตผลทางการเกษตร และทำระบบสายพานลำเลียงกับระบบคัดแยก

1.4 กิจกรรมการดำเนินงาน

ตารางที่ 1.1 ตารางแสดงขั้นตอนการดำเนินงาน

กิจกรรม	เดือน-ปี						
	มี.ค. 43	เม.ย. 43	พ.ค. 43	มิ.ย. 43	ก.ค. 43	ส.ค. 43	ก.ย. 43
ศึกษาทฤษฎีทางด้าน Image Process และจัดหาอุปกรณ์ที่ต้องใช้	←	—	—	→			
ศึกษาการเชื่อมต่อระหว่าง Computer กับ VDO PC camera		←	—	→			
ศึกษาโปรแกรมที่นำมาใช้เชื่อมต่อระหว่าง Computer กับ VDO PC camera		←	—	→			
ประกอบระบบลำเลียงและระบบคัดแยก	←	—	—	→			
ศึกษา Algorithm ในการวิเคราะห์ภาพแบบต่าง ๆ		←	—	→			
ประยุกต์และพัฒนาแนวคิดที่จะนำมาใช้วิเคราะห์วัตถุเป้าหมาย			←	—	—	→	
ทดสอบ Algorithm กับวัตถุเป้าหมายและเพิ่มฟังก์ชันการใช้งาน				←	—	—	→
ประกอบระบบการวิเคราะห์, ระบบการลำเลียง,ระบบการคัดแยกเข้าด้วยกัน และทดสอบใช้งานจริง					←	—	→

1.5 ผลที่คาดว่าจะได้รับ

- ได้รับความรู้ในเรื่องของกระบวนการวิเคราะห์ภาพ และประสบการณ์ในการทำงานจริง
- เป็นต้นแบบที่นำไปพัฒนาต่อเพื่อใช้ในระบบการผลิตในอุตสาหกรรมได้
- ทำให้เกิดแนวคิดพัฒนาที่เกิดจากคนไทย

1.6 งบประมาณที่ต้องใช้

ค่าวัสดุและค่าอุปกรณ์ รวมเป็นเงินประมาณ 7000 บาท

บทที่2
หลักการและทฤษฎี

2.1 หลักการทางเครื่องกล

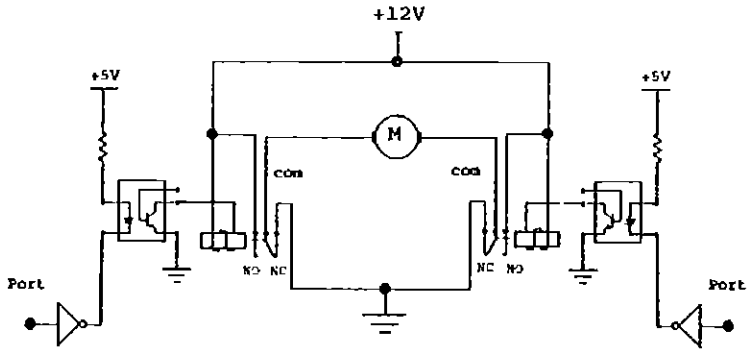
ในโครงงานนี้มีระบบลำเลียงสายพาน และระบบคัดแยก เป็นส่วนประกอบสำคัญของชุดคัดแยกผลผลิตทางการเกษตร ซึ่งมีหลักการทางเครื่องกลที่สำคัญคือ

- ระบบสายพานลำเลียง ซึ่งขับเคลื่อนด้วยมอเตอร์ที่ใช้ระบบเฟืองและโซ่เป็นตัวขับสายพานโดยจะมีแกนหมุน(Roller) อยู่ 3 ชุด
- โครงเหล็กใช้ประกอบระบบสายพาน ระบบคัดแยก และทางลำเลียงผลผลิตทางการเกษตรลงตระกร้า
- ระบบคัดแยก ซึ่งหลักการทำงานจะเป็นการผลักวัตถุ โดยใช้หลักการเดียวกับการทำงานของลูกสูบรถมอเตอร์ไซค์ ด้วยการใช้หลักว่ามอเตอร์หมุนบังคับให้แกนชุดผลักวัตถุไปผลักวัตถุและดึงชุดผลักกลับเข้าแกนของชุดผลัก โดยมีการติดชุด analog switch เพื่อควบคุมการผลักและการดึงชุดผลักกลับด้วยการส่งผลไปบอกมอเตอร์ที่บังคับชุดผลักอีกที

2.2 ระบบควบคุมมอเตอร์

การควบคุมมอเตอร์จะใช้วงจรควบคุมมอเตอร์(driver) เป็นตัวควบคุม ซึ่งในโครงงานนี้เราใช้ Driver ทิศทางเดียว

ลักษณะวงจร



รูปที่ 2.1 ภาพแสดงวงจรdrivermotor

หลักการทำงาน

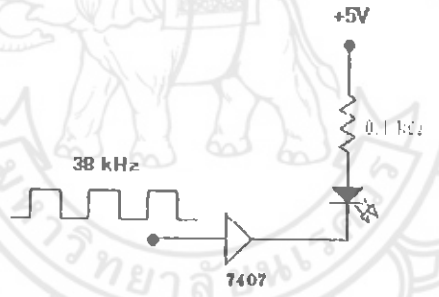
ใช้การสั่งงาน driver ควบคุมมอเตอร์ ผ่านทางบอร์ดไมโครคอนโทรลเลอร์ เพื่อสั่งให้มอเตอร์หมุน โดยในวงจรจะมีไอซีเบอร์4n25(opto-isolator) เพื่อแยกห่างทางไฟฟ้า,มี Relay 2 ตัว เพื่อการบังคับทิศทางการหมุนของมอเตอร์ให้มอเตอร์ทำงานตามต้องการ โดยเมื่อ Relay ตัวที่ 1 อยู่ในสถานะ ON นั้น Relay ตัวที่ 2 จะมีสถานะ OFF แต่ถ้า Relay ตัวที่ 1 อยู่ในสถานะ OFF นั้น Relay ตัวที่ 2 จะมีสถานะ ON

2.3 ระบบ sensor แสงอินฟราเรด

ระบบ sensor ที่ใช้ในโครงงานชิ้นนี้จะแบ่งออกเป็น 3 ส่วนด้วยกันคือ

1.ภาคส่ง

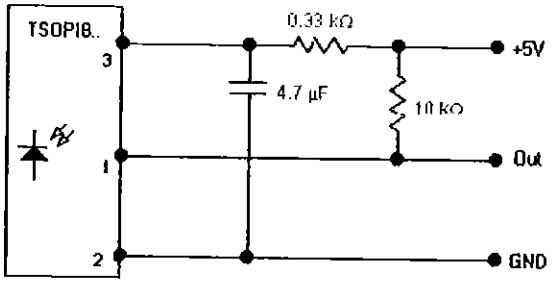
ลักษณะวงจร



รูปที่2.2 แสดงภาพวงจรภาครับของsensor

2.ภาครับ

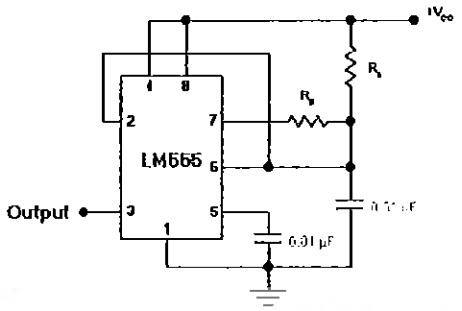
ลักษณะวงจร



รูปที่2.3 ภาพแสดงวงจรภาครับของsensor

3.<u>ชุดควบคุมความถี่</u>

<u>ลักษณะวงจร</u>



รูปที่2.4 ภาพแสดงวงจรของชุดควบคุมความถี่ให้กับsensor

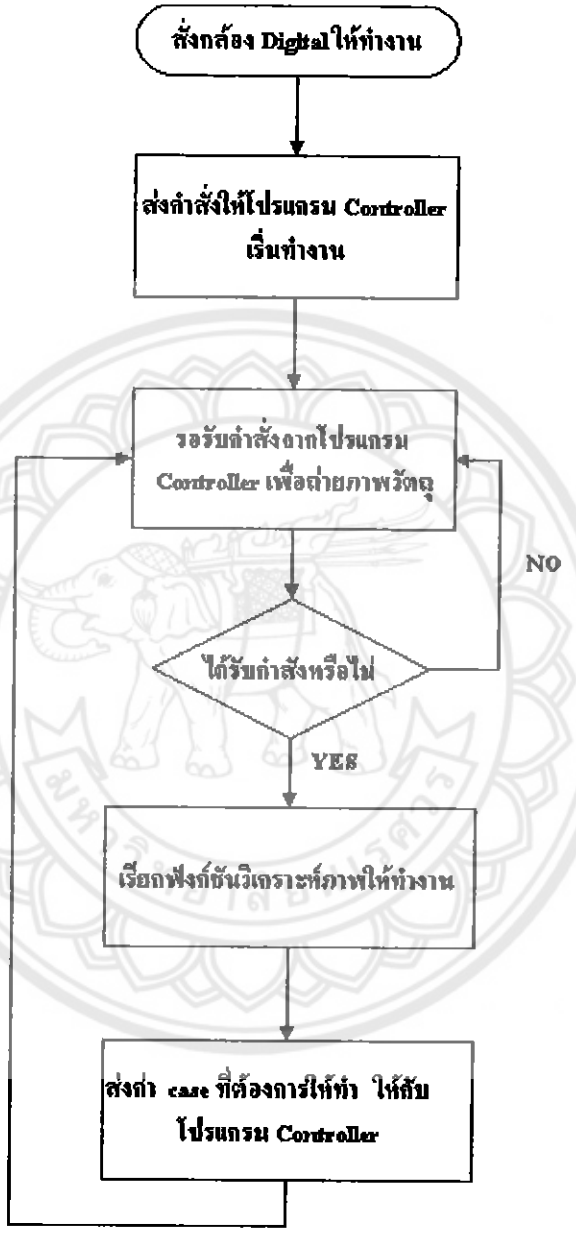
<u>หลักการทำงานของระบบ sensor</u>

ภาคส่งนั้นตัวส่งจะใช้ตัวส่งอินฟราเรด ซึ่งจะปล่งแสงตลอดเวลาดังนั้นในเวลาปกติที่ไม่มีวัตถุมาบังการส่งแสงนั้นตัวรับจะแสดงเอาท์พุทเป็นลอจิก 0 ตลอดเวลา เมื่อมีวัตถุมาบังจึงจะทำให้เกิดการสะท้อนแสงไปให้ตัวรับได้รับแสงจึงจะเกิดลอจิก 1 ส่วนภาครับนั้น จะใช้ TSOP1838 ซึ่งเป็นผลิตภัณฑ์ภาครับสำหรับอินฟราเรดรีโมทคอนโทรล ของบริษัท TEMIC ผลิตภัณฑ์ในรุ่นนี้จะมีการระบุความถี่ที่ตัวผลิตภัณฑ์ต้องการมาด้วย ดังนั้นการทำงานของภาครับจึงมีช่วงของความถี่เข้ามาเกี่ยวข้องด้วย ระบบการทำงานของ sensor ในโครงงานนี้จึงต้องมีชุดควบคุมความถี่ให้ได้ตามที่ภาครับสามารถทำงานได้ ภาครับที่ใช้นั้นสามารถทำงานได้ที่ความถี่ 38 kHz ดังนั้นเราจึงใช้ ตัวไอซี Timer 555 มาทำเป็นวงจรกำเนิดความถี่ให้ได้ตามที่ต้องการ และเป็นการควบคุมให้ได้ความถี่ในช่วงที่ภาครับสามารถทำงานได้ การแจ้งผลในการตรวจจับของระบบ sensor นี้ก็คือ เมื่อมีวัตถุมาบังตำแหน่งที่ภาคส่งส่งแสงมาเพื่อให้เกิดแสงสะท้อนไปยังภาครับ ก็จะมีเอาท์พุทออกเป็นลอจิก 1 ซึ่งจะแสดงว่า วัตถุมาถึงที่หมายที่ให้ตรวจจับ แต่หาก เอาท์พุทที่ออกมายังคงเป็นลอจิก 0 แสดงว่าวัตถุยังมาไม่ถึงที่ตรวจจับ

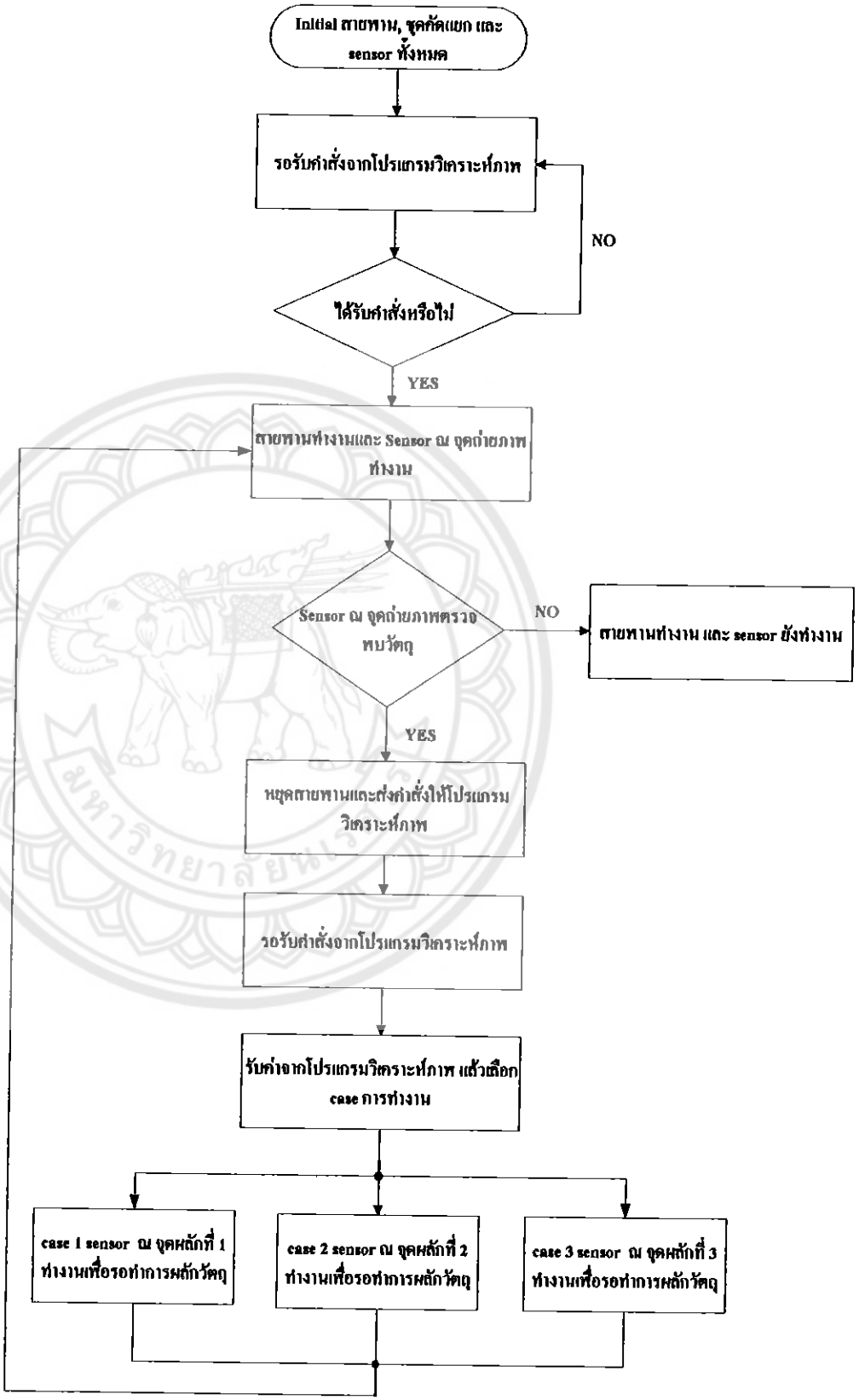
2.4 ชุดควบคุมการกลับสู่ตำแหน่งเริ่มต้นของชุดผลัก

ชุดควบคุมนี้ จะติดตั้งลิมิตสวิทย์ไว้ที่เครื่องคัดแยกตรงบริเวณขาผลักของชุดผลักทั้งสามชุด และต่อลิมิตสวิทย์แต่ละตัวเข้ากับชุดที่ควบคุมมอเตอร์ของชุดผลักแต่ละชุด การทำงานคือเมื่อชุดผลักได้ผลักวัตถุตามที่โปรแกรมประมวลผลส่งมาแล้วก็จะถอยกลับเข้าที่โดยเมื่อปลายขาผลักของชุดผลักถอยกลับมาโดนลิมิตสวิทย์ ก็จะสั่งให้ตัดไปเพื่อให้มอเตอร์หยุดหมุนและให้ชุดผลักอยู่ในตำแหน่งจุดเริ่มต้น เพื่อพร้อมรับคำสั่งต่อไปจากโปรแกรมที่ประมวลผล

2.5 การทำงานของชุดคัดแยก



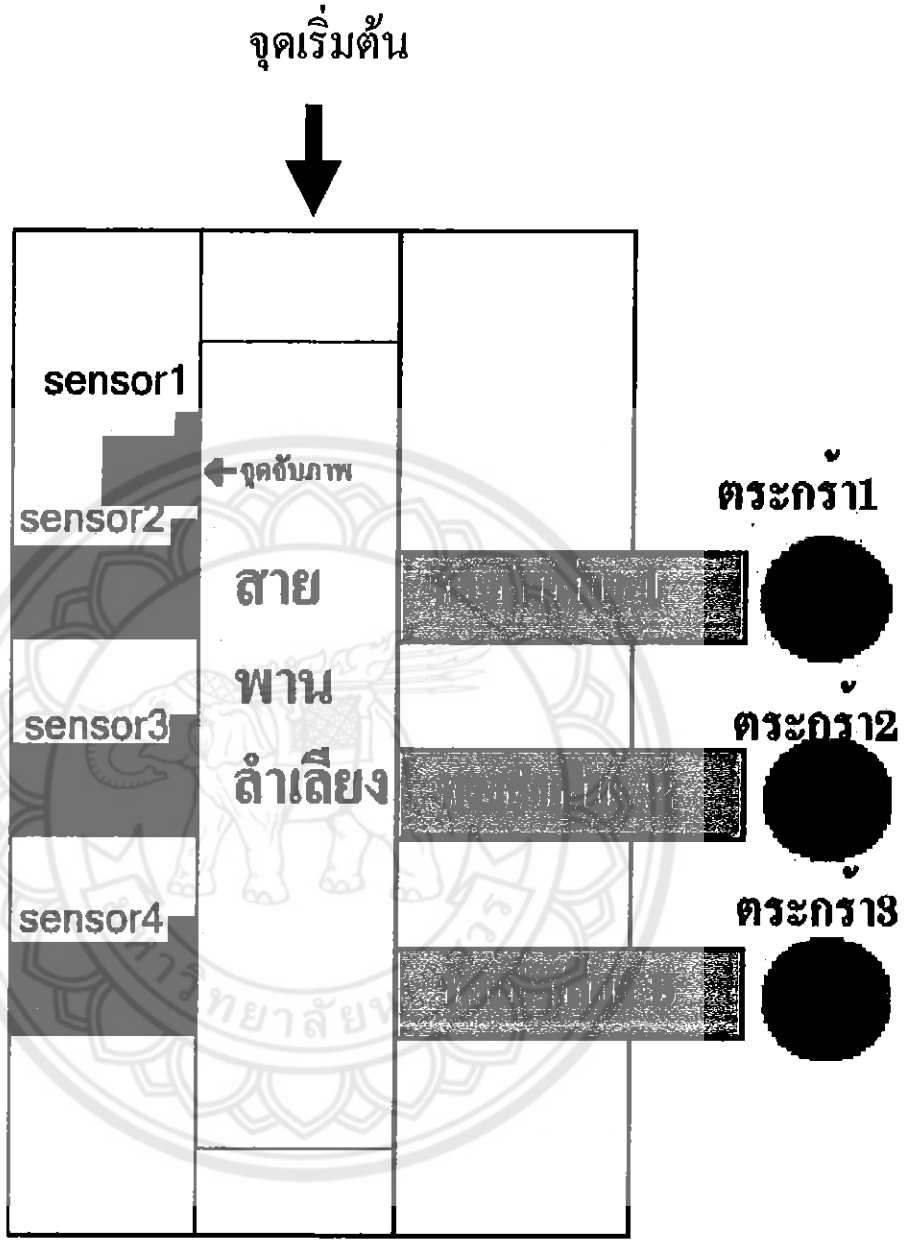
รูปที่ 2.5 ภาพแสดงผังกระบวนการทำงานของโปรแกรมวิเคราะห์ภาพ



รูปที่2.6 ภาพแสดงผังการทำงานของโปรแกรมController

คำอธิบายการทำงานของชุดคัดแยก

ชุดคัดแยกเมื่อเริ่มทำงานสายพานจะเลื่อนผลมะม่วงเข้ามาหาsensorตัวแรก เมื่อsensorตัวแรกตรวจจับผลมะม่วงได้ ก็จะสั่งมอเตอร์ของระบบสายพานหยุด เพื่อให้กล้องวีดีโอพีซีทำงานด้วยการถ่ายภาพของผลมะม่วงไว้แล้วนำเข้าประมวลผล เมื่อโปรแกรมประมวลผลเสร็จก็จะสั่งมอเตอร์ของระบบสายพานเพื่อสั่งให้สายพานเลื่อนต่อไป เพื่อนำผลมะม่วงเข้าสู่ระบบคัดแยกตามผลที่ได้จากการประมวลข้างต้น ในระบบคัดแยกนั้นจะมี sensor ตรวจจับผลมะม่วงอยู่ประจำแต่ละชุดย่อยของระบบคัดแยกซึ่งมีอยู่ 3 ชุด ด้วยกัน หากการประมวลผลแจ้งว่าผลมะม่วงที่ได้ประมวลนั้นจัดอยู่ในกลุ่ม 2 ก็จะรับผลของ sensor ตัวที่ 3 เท่านั้น เมื่อ sensor ตัวที่ 3 ตรวจจับผลมะม่วงได้นั้นก็จะสั่งให้มอเตอร์ของระบบคัดแยกของกลุ่มที่ 2 ทำงานโดยการผลักผลมะม่วงลงตระกร้าใบที่2 เมื่อเสร็จแล้วก็จะสั่งให้ดึงตัวผลักผลมะม่วงกลับเพราะตัว sensor ตรวจจับไม่พบผลมะม่วงแล้วแสดงว่าได้ทำการผลักผลมะม่วงลงตระกร้าเรียบร้อยแล้ว จากนั้นก็เริ่มทำงานใหม่ตั้งแต่ต้นต่อไปโดยทำการบันทึกจำนวนผลมะม่วงที่ได้รับการคัดแยกในแต่ละตระกร้าว่าได้จำนวนเท่าไหร่แล้วนำผลที่นับได้แจ้งกลับมาที่หน้าจอเพื่อให้ผู้ใช้ทราบ โดยจะแสดงตำแหน่งการทำงานแต่ละจุดด้วยรูปในหน้าต่อไป



รูปที่2.7 ภาพแสดงตำแหน่ง ณ.จุดต่างๆของเครื่องคัดแยก

2.6 การสื่อสารข้อมูลอนุกรม

การสื่อสารข้อมูลอนุกรมเป็นการรับหรือส่งข้อมูลในลักษณะของบิตหรือกลุ่มของบิตคราวละหนึ่งบิตเป็นลำดับเรื่อยไปจนสิ้นสุด การสื่อสารแบบนี้จะมีข้อมูลแตกต่างจากการสื่อสารแบบขนานเป็นอย่างมาก เนื่องจากข้อมูลมีการโอนย้ายมาพร้อมกันจึงมีความจำเป็นต้องใช้จำนวนเส้นสัญญาณมากขึ้นตามจำนวนบิตของข้อมูลด้วย ในขณะที่การสื่อสารแบบอนุกรมนั้นต้องการเส้นสัญญาณเพียงสองหรือสามเส้นเท่านั้น ดังนั้นการสื่อสารแบบขนานจึงไม่เหมาะสมในการสื่อสารกับอุปกรณ์ภายนอกเป็นระยะทางไกล ๆ เพราะจะทำให้สิ้นเปลืองค่าใช้จ่ายมาก

มาตรฐาน RS-232C

เพื่อที่จะทำให้อุปกรณ์จากผู้ผลิตต่างกันทำงานร่วมกันได้ มาตรฐานหลายชนิดจึงได้รับการออกแบบขึ้น มาตรฐานที่ใช้กันกว้างขวางที่สุดคือ RS-232C ถูกประกาศในปี ค.ศ. 1969 โดย Electronic Industries Association มาตรฐาน RS-232C ที่ร่างขึ้นในตอนแรกสำหรับการเชื่อมต่อระหว่างเทอร์มินัล และโมเด็มระบุคุณลักษณะทางไฟฟ้าของวงจรระหว่างอุปกรณ์ 2 ตัวและการกำหนดชื่อและหมายเลขแก่สายที่จำเป็นสำหรับการเชื่อมต่อวงจร

ชื่อตามมาตรฐาน RS-232C จำได้ยากในทางปฏิบัติจึงใช้ชื่อย่อแทน ตัวอย่างเช่น สายเส้นที่สองมีชื่อว่า BA แต่ที่ใช้กันทั่วไปว่า TXD (Transmitted Data) ตามมาตรฐาน RS-232C สายเส้นที่สองนำข้อมูลจากเทอร์มินัลสู่โมเด็ม เพื่อให้การทำงานถูกต้องเทอร์มินัลต้องส่ง output ออกที่สายสัญญานเส้นที่ 2 และโมเด็มต้องรับข้อมูลบนสายเส้นที่สอง เพราะฉะนั้นสายเส้นที่สองเป็นสายส่งข้อมูลสำหรับอุปกรณ์บางอย่าง และสามารถเป็นสายรับข้อมูลสำหรับอุปกรณ์บางอย่าง การเชื่อมต่อโดยตรงจากสายเส้นที่ 2 บนอุปกรณ์หนึ่งเข้ากับสายเส้นที่สองของอุปกรณ์อีกตัวหนึ่ง สามารถทำได้ก็ต่อเมื่ออุปกรณ์หนึ่งส่งข้อมูลบนสายเส้นที่สอง และอีกตัวหนึ่งรับข้อมูลบนสายเส้นที่สอง

เพื่อป้องกันไม่ให้อุปกรณ์ส่งข้อมูลบนสายเส้นเดียวกัน อุปกรณ์จึงถูกแบ่งเป็น 2 ชนิด อุปกรณ์บางอย่างเช่น เทอร์มินัล ซึ่งใช้สายเส้นที่สองสำหรับเอาท์พุต เรียกว่า DTE (Data Terminal Equipment) อุปกรณ์อย่างเช่น Modem ซึ่งใช้สายเส้นที่สองสำหรับ อินพุต เรียกว่า DCD (Data Communication Equipment)

รูปแบบสัญญาณทางไฟฟ้า

มาตรฐาน RS-232C กำหนดคุณลักษณะของสัญญาณทางไฟฟ้าที่ถูกใช้ในการเชื่อมต่ออนุกรมโดยตรงมีเพียง 2 ลักษณะคือ SPACE แสดงถึง binary 0 หรือแรงดันไฟฟ้าบวก และ MARK แสดงถึง Binary 1 หรือแรงดันไฟฟ้าลบ

บนสายข้อมูล (เช่นสาย 2 และ 3) แรงดันไฟฟ้าบวกแสดงถึงค่าลอจิก 0 และแรงดันไฟฟ้าลบแสดงถึงค่าลอจิก 1 บนสาย Handshaking (เช่น DTR และ DSR) แรงดันไฟฟ้าบวกแสดงว่าส่งข้อมูลได้ ส่วนแรงดันไฟฟ้าลบหมายถึงหยุดส่งข้อมูล

แรงดันไฟฟ้าบวกสถานะ Space อยู่ระหว่าง +5 ถึง +15 โวลต์ สำหรับเอาท์พุต และระหว่าง +3 และ +15 โวลต์สำหรับอินพุต ความแตกต่างมีไว้เพื่อกรณีแรงดันไฟฟ้าสูญหายเนื่องจากความยาวของสายสัญญาณ ในทำนองเดียวกัน แรงดันไฟฟ้าลบ สถานะ MARK กำหนดระหว่าง –5 ถึง –15 โวลต์สำหรับ output และ –3 ถึง –15 โวลต์สำหรับ Input

สังเกตว่า ถ้าสายสัญญาณยาวเกินไป ระดับแรงดันไฟฟ้าจะตกลงเกินขอบเขตที่ยอมรับได้ นอกจากนี้ความจุไฟฟ้าที่เกิดขึ้นจะมีผลกับคุณภาพของสัญญาณ โดยทำให้การเปลี่ยนแปลงจากแรงดันไฟบวกเป็นไฟลบไม่ชัดเจน RS-232C ไม่ได้มุ่งหวังให้นำไปใช้สำหรับระยะทางไกล ๆ และโดยทั่วไป 50 ฟุตเป็นระยะทางที่ไกลที่สุดในการใช้สายสัญญาณปกติที่อัตราการส่งข้อมูลปกติ ถ้าอุปกรณ์อยู่ห่างกันมาก อาจจำเป็นต้องใช้โมเด็ม หรืออุปกรณ์อื่น

2.7 การอินเตอร์รัปต์ของการสื่อสารอนุกรม

เนื่องจากการส่งหรือรับข้อมูลอนุกรมในการส่งข้อมูลไบต์หนึ่ง ๆ ค่อนข้างจะใช้เวลานานหลายมิลลิวินาที ดังนั้นเพื่อให้การจัดการเกี่ยวกับการสื่อสารแบบนี้เป็นไปอย่างมีประสิทธิภาพ จึงได้กำหนดให้บิตหรือแฟล็กสถานะที่เกี่ยวข้องทั้งหมด จัดรวมอยู่ภายในรีจิสเตอร์ SCON เท่านั้น เช่น แฟล็ก TI ซึ่งจะมีค่าเป็น 1 เมื่อข้อมูลได้ทำการส่งออกไปภายนอกเสร็จสิ้นแล้ว และแฟล็ก R1 ซึ่งจะมีค่าเป็น 1 เพื่อแจ้งให้ทราบว่าได้รับข้อมูลผ่านเข้ามาทางพอร์ตอนุกรม เมื่อแฟล็กตัวใดตัวหนึ่งนี้มีค่าเป็น 1 จะมีผลทำให้เกิดการอินเตอร์รัปต์ขึ้น ดังนั้นภายในโปรแกรมจะต้องทำการตรวจสอบจากสภาวะของแฟล็กเหล่านี้เองว่ามีการอินเตอร์รัปต์ขึ้นด้วยสาเหตุใด จากนั้นจึงค่อยทำการกำหนดค่า 0 ให้กับแฟล็กนั้น

2.8 กระบวนการรับและส่งข้อมูลอนุกรม

การส่งข้อมูลออกทางพอร์ตอนุกรมของ 8031 จะเริ่มต้นขึ้น ภายหลังเมื่อมีการเขียนข้อมูลลงในรีจิสเตอร์ SBUF ซึ่ง รีจิสเตอร์ SBUF ของ CPU 8031 เป็นบัฟเฟอร์ขนาด 8 บิต สำหรับการสื่อสารข้อมูลแบบอนุกรมทั้งการรับและการส่งข้อมูล ซึ่งตามความเป็นจริงแล้วบัฟเฟอร์นี้มีอยู่ด้วยกันสองชุดและแยกจากกันอย่างชัดเจน สำหรับการส่งและการรับ โดยซีพียูจะทำการจัดการเลือกบัฟเฟอร์ที่เหมาะสมให้โดยอัตโนมัติ) ข้อมูลนี้จะถูกจัดการด้วยวิธีการทางด้านฮาร์ดแวร์ในการเลื่อนบิตและส่งสัญญาณออกไปภายนอกโดยอัตโนมัติ เมื่อข้อมูลเหล่านี้ได้ส่งออกครบถ้วยแล้ว จึง

จะทำการกำหนดค่าแฟล็ก T1 ให้เป็น 1 เพื่อแจ้งให้ทราบว่าขณะนี้รีจิสเตอร์ SBUF ว่าง และพร้อมที่จะส่งข้อมูลไบต์ต่อไปแล้ว ในกรณีที่ผู้ใช้เขียนข้อมูลใหม่ลงในรีจิสเตอร์ SBUF โดยไม่รอให้แฟล็ก T1 มีค่าเป็น 1 ก่อนจะมีผลทำให้ข้อมูลที่ส่งออกไปผิดพลาดได้

สำหรับการรับข้อมูลจากพอร์ตอนุกรมจะต้องเริ่มต้นโดยการกำหนดค่าบิต REN (Reciever Enable) ให้มีค่าเป็น 1 ก่อน หลังจากนั้นเมื่อมีบิตของข้อมูลถูกส่งเข้ามาจากภายนอกระบบฮาร์ดแวร์ของ 8051 จึงจะทำการเลื่อนบิตเหล่านี้เข้ามาโดยอัตโนมัติ และเมื่อบิตสุดท้ายถูกเลื่อนเข้ามาเรียบร้อยแล้ว ข้อมูลนั้นจะถูกย้ายมาเก็บไว้ยังรีจิสเตอร์ SBUF และทำการกำหนดให้แฟล็ก R1 ให้มีค่าเป็น 1 ซึ่งมีผลทำให้เกิดการอินเตอร์รัปต์โปรแกรมขึ้น

2.9 การทำงานของการสื่อสารทางพอร์ตอนุกรม

กำหนดให้ mode การทำงานเป็น mode 1 และกำหนดอัตราการส่งโดยตั้งค่ารีจิสเตอร์ PCON กำหนดค่าที่เกี่ยวข้องกับการอินเตอร์รัปต์โดยผ่านรีจีสเตอร์ SCON และ TMOD เลือกประเภทสัญญาณอินเตอร์รัปต์ซึ่งอยู่ในรีจีสเตอร์ TCON เมื่อกำหนดค่าให้กับรีจิสเตอร์ต่าง ๆให้กับบอร์ดคอนโทรลแล้ว บอร์ดคอนโทรลจึงพร้อมจะรับส่งข้อมูลผ่านทางพอร์ตอนุกรมได้โดยติดต่อผ่าน
รีจิสเตอร์ SBUF

การรับและส่งข้อมูลระหว่าง บอร์ดคอนโทรล และ Computer นั้นจะใช้ โปรแกรม Delphi ในการจัดการ โดยจะใช้คอมโพเนนต์ RS232 มาช่วยในการจัดการซึ่งจะมีคำสั่งในการควบคุมพอร์ตอนุกรม เช่น การกำหนดอัตราส่ง, เลือกพอร์ต (Com1 หรือ Com2), Read Port, Write Port มาให้ ซึ่งสามารถนำมาประยุกต์ใช้ได้

2.10 การติดตั้งการสื่อสารทางพอร์ตอนุกรมด้านฮาร์ดแวร์

การเชื่อมต่อระหว่างพอร์ตอนุกรมและ บอร์ดคอนโทรล จะใช้คอนเน็กเตอร์ชนิด D(D-type) ชนิด 25 ขา โดยเราใช้มาตราฐาน RS-232C โดยมีการจัดขา และรายละเอียดฟังก์ชันการทำงานดังนี้

ตารางที่ 2.1 ตารางแสดงคอนเน็กเตอร์ของพอร์ต RS-232 แบบ 9 ขา

คอนเน็กเตอร์ของพอร์ต RS-232C แบบ 9 ขา

ขาที่	ลักษณะการทำงาน	คำอธิบาย
1	Input	Data Carrier detect (DCD)
2	Input	Receive Data(RX)
3	Output	Transmit Data(TX)
4	GND	Data Terminal Ready(DTR)
5	Input	Signal Ground
6	Input	Data Set Ready (DSR)
7	Output	Request To Send (RTS)
8	Input	Clear To Send (CTS)
9	Input	Ring Indicator (RI)

การเชื่อมต่อจะเชื่อมขา 5 กับ ขา 6 หากัน และ ขา 7 และ ขา 8 หากันซึ่งจะทำภายในคอนเน็กเตอร์ ส่วนขาที่จะเชื่อมต่อไปหาบอร์ดคอนโทรลมีด้วยกัน 3 ขา คือ ขา RX, ขา TX และขากราวด์ โดยขา RX จากพอร์ตอนุกรมนั้นจะเชื่อมต่อไปยังขา TX ของบอร์ดคอนโทรล และขา TX จากพอร์อนุกรมก็จะเชื่อมต่อไปยังขา RX ของบอร์ดคอนโทรล และขากราวด์กับขากราวด์

2.11 Microcontroller 8031

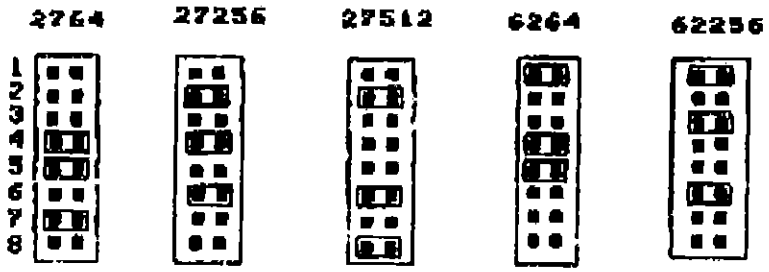
8031 เป็นไมโครคอนโทรลเลอร์ที่มีหน่วยการทำงานพื้นฐานประกอบไปด้วยหน่วยประมวลผลกลาง (CPU) ขนาด 8 บิต หน่วยประมวลผลสำหรับข้อมูลแบบ(Boolean Processor) ความสามารถในการอ้างตำแหน่งของหน่วยความจำโปรแกรม ROM 64 Kbyte ความสามารถในการอ้างตำแหน่งหน่วยความจำข้อมูล RAM 64 Kbyte หน่วยความจำแบบแรมภายในจำนวน 128 byte พอร์ต อินพุต/เอาท์พุต แบบขนานจำนวน 32 บิต ซึ่งสามารถแยกการทางานได้อย่างอิสระ วงจรนับจับเวลา ขนาด 16 บิต จำนวนสองวงจร วงจรการสื่อสารแบบอนุกรมแบบฟูลดูเพล็กซ์ (Full Duplex) วงจรควบคุมการอินเตอร์รัปต์จากแหล่งกำเนิดสัญญาณ 6 ประเภท พร้อมการกำหนดลำดับความสำคัญได้สองระดับ วงจรออสซิเลเตอร์ภายในการทำงานสามารถทำงานโดยเป็นพอร์ตอินพุต เอาต์พุตธรรมดา หรือเป็นแอสเดรสบัสและดาตาบัสก็ได้ขึ้นอยู่กับการเลือกใช้งาน

การทำงานของ บอร์ดคอนโทรล

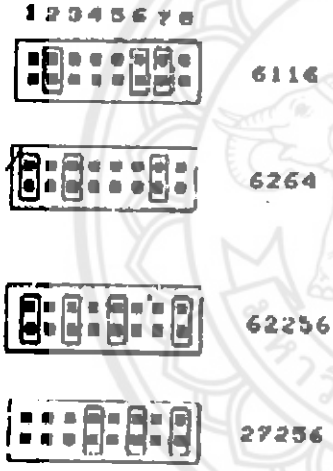
บอร์ดคอนโทรล จะใช้บอร์ดของ ETT รุ่น CP-SB31 ซึ่งใช้ CPU8031 การกำหนดค่า ต่าง ๆ ให้กับ ตัวบอร์ด กำหนดโดยวิธีการกำหนดจั้มเปอร์ และกำหนดค่าภายในโปรแกรมที่จะสั่งงานบอร์ดอีกให้เข้ากัน เนื่องจาก CP-SB31 ถูกสร้างมาให้เป็นอิสระในการเลือกใช้หน่วยความจำได้หลายขนาดทั้ง EPROM และ RAM รวมทั้งตำแหน่งแอดเดรสของหน่วยความจำผู้ใช้ก็ยังสามารถกำหนดได้ตามต้องการซึ่งทั้งหมดนี้ขึ้นอยู่กับการใส่ตำแหน่งของ JUMPER ต่าง ๆ ให้ถูกต้อง ซึ่ง U3 และ U4 จะถูกควบคุมด้วย JUMPER ซึ่งมีรายละเอียดดังนี้

U3	JP3	เลือกเบอร์ของชิปหน่วยความจำที่ใส่อยู่บน U3 (2764, 27256, 27512, 6264, 62256)
	JP4	สำหรับเลือกว่าจะให้หน่วยความจำที่ U3 เป็น DATA MEMORY หรือ CODE MEMORY หรือเป็นทั้ง CODE และ DATA MEMORY
	JP7	เลือกตำแหน่งเริ่มต้นและขนาดของหน่วยความจำ U3
U4	JP5	เลือกเบอร์ของชิปหน่วยความจำที่ใส่อยู่บน U4 (27256,6116,6264,62256)
	JP6	เลือกลักษณะการทำงานของ U4 ว่าจะให้เป็น DATA MEMORY หรือ CODE MEMORY หรือเป็นทั้ง DATA และ CODE MEMORY
	JP8	เลือกตำแหน่งเริ่มต้นและขนาดของหน่วยความจำ U4
	JP9	เลือกว่าจะอนุญาติให้มีการใช้ I/O พอร์ต (8255) ภายนอกอีกหรือไม่ ถ้าไม่มีพอร์ตภายนอก U4 จะมีขนาดสูงสุดได้ถึง 32 KB

JP3 (สำหรับ U3) กำหนดเป็น 27256



JP5(สำหรับ U4) กำหนดเป็น 6264



JP4&JP6 (เลือกลักษณะของ U3 และ U4) กำหนดเป็น CODE และ DATA ตามลำดับ

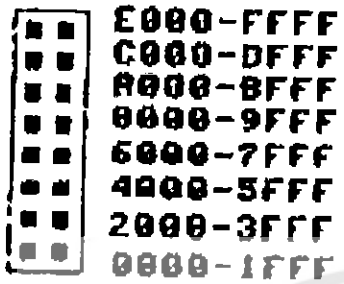
COMBINE CODE DATA

DATA = DATA MEMORY

CODE = CODE MENORY

COMBINE = DATA AND CODE MENORY

JP7&JP8 (เลือกตำแหน่งเริ่มต้นและขนาดของหน่วยความจำ) แต่ละจุดของ JUMP ที่ใส่ลงไปจะมีขนาดของหน่วยความจำ 8 K เพิ่มขึ้นจากจุดเริ่มต้น



JP9 กำหนดให้เป็น PORT

PORT 8255 DECODE	PORT A = E0E0H
	PORT B = E0E1H
	PORT C = E0E2H
PCON = CONTROL PORT = E0E3H	

การทำงานของ EPROM EMULATOR

EPROM EMULATOR ใช้บอร์ดของ ETT รุ่น ET-EM PLUS ซึ่งทำงานแทนส่วนของตัว EPROM หรือ RAM ซึ่งจะมีความง่ายและสะดวกกว่าการใช้ EPROM หรือ RAM จริง ๆ ทำให้เหมาะกับการพัฒนาระบบไมโครต่าง ๆ หรือในการแก้ไขเปลี่ยนแปลงโปรแกรมการทำงานของเครื่องเป็นไปโดยง่าย

เราสามารถเขียนข้อมูลเข้าไปในส่วนของ ET-EM PLUS ได้โดยตรงซึ่งต่างกับการที่เราต้องนำ EPROM นั้นออกมาเขียนเปลี่ยนแปลงแล้วจึงนำกลับเข้าไปใส่ในวงจรใหม่

งานที่นำไปใช้

ET-EM PLUS สามารถนำไปใช้งานได้หลายรูปแบบขึ้นอยู่กับผู้ใช้จะนำไปใช้งาน เช่น

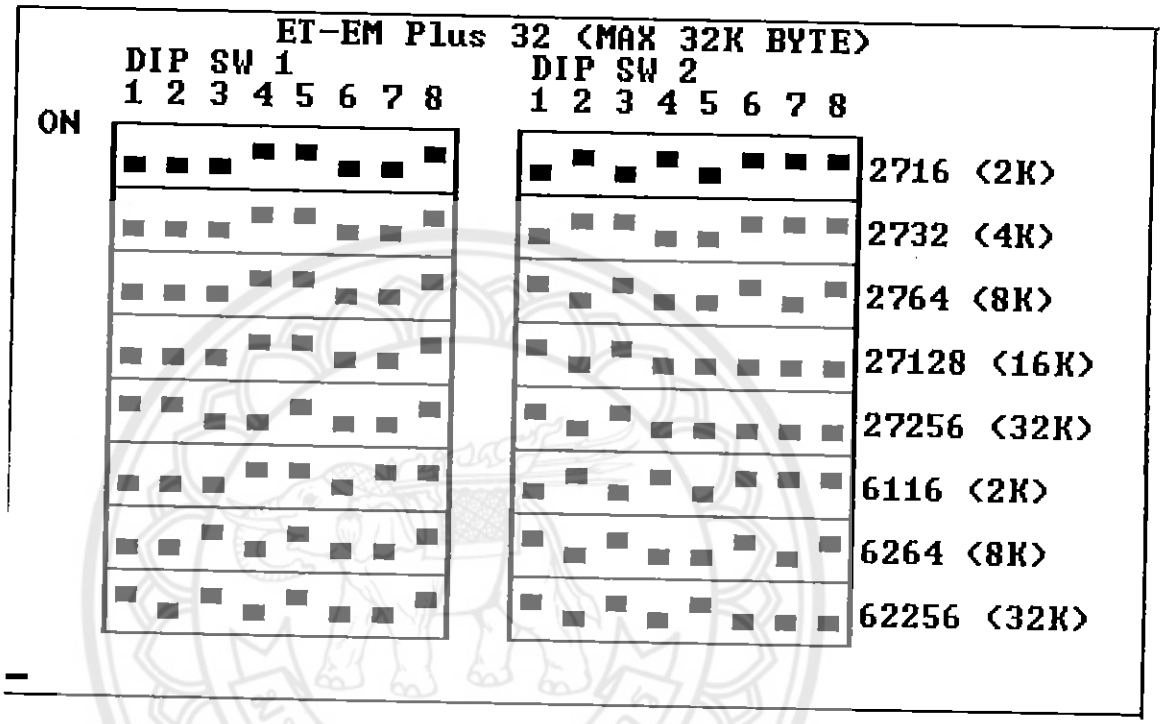
1. ROM MONITOR ใช้ในการเขียนโปรแกรมในส่วนของ MONOTOR PROGRAM ของเครื่องไมโครต่าง ๆ เพราะในการเริ่มต้นพัฒนาระบบไมโครอะไรก็ตามไมโคร นั้นยังไม่มีส่วนของโปรแกรมในการเขียนอ่านข้อมูลจากภายนอกซึ่งถ้านำ ET-EM PLUS มาใช้ก็จะสะดวกกว่าใช้ EPROM เป็น MONITOR จริง ๆ มาก
2. CHARACTER GENERATOR เราสามารถใช้ ET-EM PLUS ในการทำเป็น CHARACTER GENERATOR ในการเปลี่ยนแปลงหรือเพิ่มเติมตัวอักษรเข้าไป เช่น ทำภาษาไทยในเครื่องพิมพ์ ซึ่งเราสามารถทดสอบเขียนแล้วทดสอบทำงานได้โดยตรง
3. BIOS เป็น ROM BIOS ในเครื่อง PC ใช้ทดสอบหรือ DEBUG ตัวโปรแกรม BIOS ของเครื่องหรือใช้ทดสอบเขียนโปรแกรมตรวจสอบการทำงานเป็นส่วน ๆ ของเครื่องได้ ฯลฯ

คุณสมบัติของ ET-EM PLUS

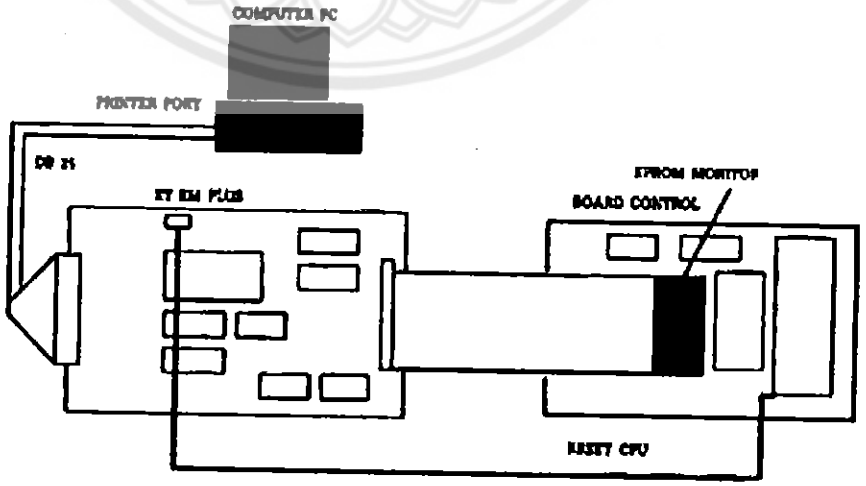
1. สามารถส่งผ่านข้อมูลจากเครื่อง PC ได้ทาง PRINTER PORT ทำให้ง่ายในการจะหาเครื่อง PC มาใช้เพราะเครื่อง PC ต่าง ๆ นั้นจะมีส่วนของ PRINTER PORT อยู่แล้ว
2. ใช้กับ FILE ได้หลายรูปแบบ เช่น BINARY FILE, INTEL HEX FILE, MOTOLORA FILE (S FORMAT)
3. สามารถเปลี่ยนแปลงข้อมูลใน ET-EM PLUS ทั้งหมดหรือบาง BYTE ก็ได้โดยไม่จำเป็นต้อง LOAD FILE ใหม่หมด
4. สามารถตั้ง OFFSET คือค่าตำแหน่งที่อยู่ของข้อมูลจาก FILE ได้โดยตรงอย่างอิสระ
5. สามารถเลือกรูปแบบการส่งข้อมูลจาก FILE ได้หลายแบบ เช่น ส่งเต็มข้อมูล,ส่งข้อมูลเฉพาะ BYTE คู่, ส่งข้อมูลเฉพาะ BYTE คี่ ซึ่งทำให้เราสามารถนำ ET-EM PLUS 2 ตัวมาต่อกับ PRINTER PORT 2 ชุด ใช้พัฒนาระบบเครื่องที่ จำเป็นต้องใช้ 16 BIT DATA หรือที่ต้องใช้ EPROM 2 ตัว ของเครื่อง 16 BIT ที่ใช้เป็น BIOS ของเครื่อง
6. สามารถต่อกับ PRINTER PORT ได้ทั้ง LPT1, LPT2 หรือ MONO CARD ได้โดยเป็นอิสระต่อกัน
7. มีความเร็วในการส่งผ่านข้อมูลสูงมาก
8. สามารถ RESET CPU ได้เมื่อ LOAD ข้อมูล แล้ว โดยมี แบบ RESTE LOW และ RESET HIGH

การนำไปใช้งาน

ปรับ DIP SWITCH ของ ET-EMPLUS ให้ตรงกับส่วนของวงจรที่จะนำไปใช้งานดังรูป โดยจะกำหนดตามค่าของ RAM ที่ใช้คือ 32 KBYTE (เบอร์ 27256)



การต่อเข้ากับคอมพิวเตอร์โดยต่อเข้าทาง PRINTER PORT โดยใช้สาย DB25 PIN ดังรูป



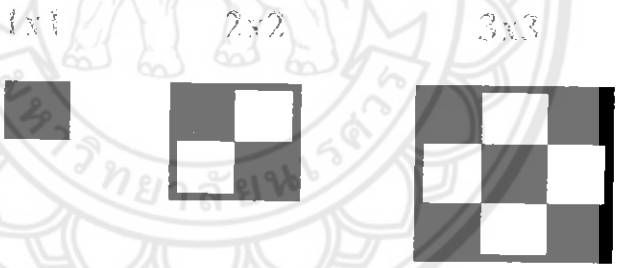
รูปที่ 2.8 ภาพแสดงการต่อเข้ากับคอมพิวเตอร์โดยต่อเข้าทาง PRINTER PORT โดยใช้สาย DB25 PIN

2.12 รูปแบบข้อมูลภาพโดยทั่วไป

รูปแบบที่มีใช้อยู่โดยทั่วไปมีรูปแบบดังนี้คือ GIF IFF/LBM WPG BMP PIC TGA PCX และ TIFF เป็นต้น สำหรับรูปแบบใหญ่ ๆ ที่นิยมใช้กันมากคือ PCX GIF BMP และ TIFF แต่รูปแบบหนึ่งที่ได้รับความนิยมคือ BMP หรือ บิตแมพ สาเหตุก็เพราะว่าข้อมูลแบบบิตแมพนั้นไม่มีการบีบข้อมูล (COMPRESSION) จึงมีความสะดวกในการอ่านข้อมูลอย่างมากและเป็นรูปแบบที่ละเอียดสามารถประยุกต์ เปลี่ยนแปลงไปเป็นไฟล์รูปแบบอื่นได้ และเป็นรูปแบบข้อมูลกลางที่ใช้ในการแลกเปลี่ยนระหว่างกันของโปรแกรมประยุกต์ต่าง ๆ ใช้ในการส่งข้อมูลไปให้อุปกรณ์ต่าง ๆ เช่น อุปกรณ์การแสดงผล เป็นต้น แต่ในรูปแบบบิตแมพนี้ต้องเปลืองเนื้อที่ในการเก็บข้อมูลมากกว่ารูปแบบอื่นที่ใช้การเข้ารหัสบีบข้อมูล

เรื่องของกระบวนการวิเคราะห์ภาพ

ภาพที่กล้องดิจิตอลบันทึกไว้นั้นเป็น.BMP หรือ Bitmap File ซึ่งภาพที่ได้จะถูกเก็บเป็นพื้นที่สี่เหลี่ยมเล็กๆที่เรียกว่า pixel ซึ่ง pixel นั้นเป็นองค์ประกอบที่เล็กที่สุดของภาพ



โดยแต่ละ pixel จะถูกระบุตำแหน่งโดยแกนพิกัดในระนาบ 2 มิติ คือ แกน x และ y โดยปกติเราจะให้จุดกำเนิดของแกนพิกัดจะอยู่ทางมุมซ้ายของภาพ

ภาพที่ได้จากกล้องดิจิตอลของเรานั้นจะสามารถมีความละเอียดได้สูงสุด 640 * 480 pixel และความละเอียด (Resolution ที่ 72 pixel ตัวนี้) ซึ่งหมายความว่า ในพื้นที่ 1 ตารางนิ้วจะมีจำนวน pixel เท่ากับ 72 * 72 = 5184 pixel ถ้าหา Resolution สูงจะแสดงรายละเอียดของสีได้ดี ทำให้ภาพมีคุณภาพมากยิ่งขึ้น

การนำกล้องดิจิตอลมาใช้งาน

เราจะเรียกใช้งาน WINDOW API (Application Program Interface) ซึ่งก็คือฟังก์ชันต่างที่ window เตรียมไว้ให้เราใช้งาน โดยฟังก์ชันต่างๆ เหล่านั้นจะถูกเก็บใน File .DLL ของ windows โดยเราจะเรียกใช้งานจาก Avicap.dll และ Avicap32.dll ซึ่งเราจะใช้งานผ่านยูนิต Vidcocap ซึ่งฟังก์ชันให้เราเรียกใช้งานได้สะดวก ไม่ว่าจะเป็นการ set ความละเอียด, ขนาด, ปรับแต่งแสง, สี, ความเข้ม

2.13 Algorithm ในการวิเคราะห์ภาพ

หลักการที่ใช้ในการวิเคราะห์ภาพนั้นจะแบ่งเป็น 2 ส่วนด้วยกันคือ การวิเคราะห์ในส่วนของขนาดวัตถุ และการวิเคราะห์ในส่วนของสีของวัตถุ

ในการวิเคราะห์หาขนาดของวัตถุนั้นจะวิเคราะห์โดยการอ่านค่าสีในแต่ละ pixel ของภาพที่ถ่ายมา โดยภาพที่ถ่ายมานั้นจะถูกบันทึกให้อยู่ใน mode RGB Mode ซึ่งมีการกำหนดค่าความเข้มของสีแบ่งเป็น R = แดง G = เขียว B = น้ำเงิน โดยค่าของสีที่ได้จากการวิเคราะห์นั้นเป็นเลขฐาน 16 ตัวอย่างเช่น สีแดง = FF0000 สีเขียว = 00FF00 สีน้ำเงิน = 0000FF ซึ่งแต่ละสีจะมี 256 ระดับ เมื่อรวมทั้งหมดแล้ว ภาพที่เกิดจาก RGB Mode จะมี 16.7 ล้านสี การวิเคราะห์จึงวิเคราะห์หาสีที่มีอยู่ในแต่ละ pixel หาก pixel วิเคราะห์แล้วพบค่าสีที่ได้นั้นมากกว่าค่าที่ใช้เทียบ ก็จะไม่นับ pixel นั้น แต่หากค่าสีที่ได้น้อยกว่าก็จะนับ pixel นั้น และจากนั้นก็จะนำผลการวิเคราะห์นั้นมาสร้างเป็นภาพซึ่งมีจุดสีดำและขาว เพื่อแสดงผลในการบอกขนาดของวัตถุ โดยภาพนี้จะมีขนาดเท่ากับภาพเดิม จากที่วิเคราะห์ได้ผลมานั้น ก็จะสามารถนำไปแบ่งแยกขนาดของวัตถุได้โดยใช้เท่ากับค่าที่ได้ตั้งไว้โดยมีอยู่ 3 ช่วงของขนาด แล้วแต่เรากำหนด

ในส่วนของการวิเคราะห์หาความสุกดิบของวัตถุนั้น จะนำค่าสีแดง สีเขียว สีน้ำเงิน ที่มีอยู่ในแต่ pixel มาเก็บไว้และนำค่าทั้งหมดที่นับไว้มาหาค่าเฉลี่ยของสีโดยรวมทั้งหมดของภาพ และนำไปแสดงผล หากสีที่ได้มาจากการเฉลี่ยนั้นใกล้เคียงกับสีเหลืองก็จะถือว่าอยู่ในเกณฑ์สุกได้ ซึ่งภาพที่ถ่ายมานั้นจะเป็นภาพวัตถุวางบนพื้นสีขาว เพราะฉะนั้นภาพที่ได้มาก็จะเห็นเพียงวัตถุอย่างชัดเจน

บทที่ 3
วิธีการดำเนินงาน

3.1 การดำเนินโครงงาน

การดำเนินโครงงานนี้เริ่มต้นด้วยการศึกษารายละเอียดเกี่ยวกับตัวชิ้นงาน โดยเริ่มศึกษาจากหลักทฤษฎีและความรู้เกี่ยวกับด้านต่างของชิ้นงานที่จะทำเช่น ความรู้ทางด้านsensor, ความรู้ทางด้าน drivermotor เป็นต้น จากนั้นก็ทำการออกแบบตัวเครื่องคัดแยก และจัดหาวัสดุที่จะนำมาทำตัวเครื่อง ต่อมาก็จัดทำตัวเครื่อง พร้อมกับศึกษาทฤษฎีทางด้านการวิเคราะห์ภาพและประกอบชุดควบคุมด้านต่างๆ เมื่อทำตัวเครื่องเสร็จก็นำชุดควบคุมมาประกอบเข้ากับตัวเครื่องพร้อมกับทดลองควบคู่ไปกับการพัฒนาโปรแกรม ก่อนจะนำไปทดลองและเก็บผลการทำงานของเครื่องคัดแยกที่ได้จากการทดลอง ซึ่งการดำเนินงานสามารถสรุปเป็นขั้นตอนได้ดังต่อไปนี้

1. ศึกษารายละเอียดและหลักการ ความรู้ ทฤษฎีต่างๆ ที่เกี่ยวข้องกับชิ้นงาน
2. ออกแบบตัวเครื่องคัดแยกพร้อมกับจัดหาและเลือกวัสดุที่จะนำมาประกอบตัวเครื่อง
3. ประกอบตัวเครื่องคัดแยกตามแบบที่จัดทำขึ้นรวมทั้งจัดทำบอร์ดควบคุมต่างๆที่ต้องใช้ในตัวโครงงาน
4. ทดลองการทำงานของบอร์ดควบคุมต่างๆที่จัดทำขึ้นก่อนทำการติดตั้งกับตัวเครื่องคัดแยก
5. ทำการติดตั้งชุดบอร์ดควบคุมต่างๆเข้ากับตัวเครื่องคัดแยก
6. ทำการทดลองการส่งข้อมูลระหว่างตัวโปรแกรมกับเครื่องคัดแยก
7. ทำการพัฒนาโปรแกรมพร้อมไปกับการทดลองขั้นต้น
8. เมื่อโปรแกรมถูกพัฒนาเสร็จแล้วก็ทำการทดลองจริงและเก็บผลการทดลอง
9. ตรวจสอบผลการทดลองหาข้อผิดพลาดเพื่อทำการแก้ไข
10. ทดลองและเก็บผลการทดลองจริงอีกครั้งหลังการแก้ไขข้อผิดพลาดที่ตรวจพบ

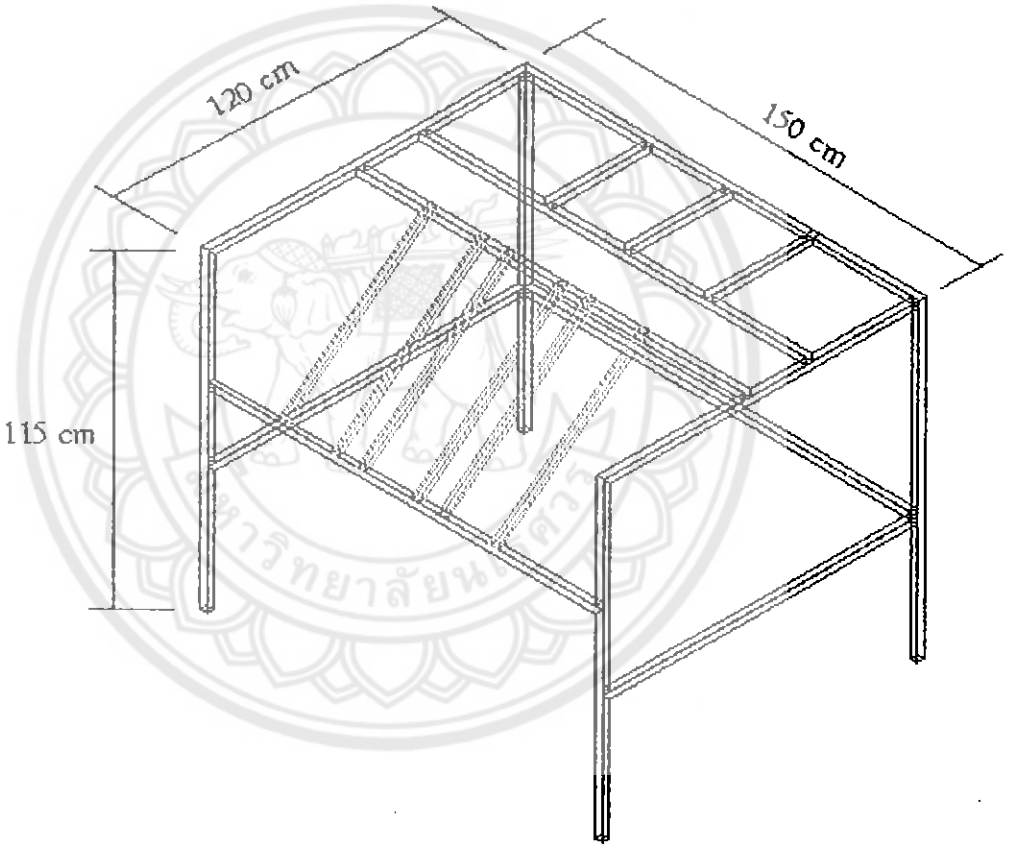
3.2 การออกแบบ

ขั้นตอนการออกแบบแบ่งเป็น 3 ส่วนด้วยกัน คือ ส่วนตัวเครื่องคัดแยก ส่วนชุดควบคุมต่างๆ และส่วนที่ติดต่อกับผู้ใช้

3.2.1 การออกแบบส่วนตัวเครื่องคัดแยก

การออกแบบตัวเครื่องมีขั้นตอนดังต่อไปนี้

1. ออกแบบโครงเหล็กของเครื่องคัดแยก



รูปที่ 3.1 ภาพแสดงแบบโครงสร้างของเครื่องคัดแยก

2. จัดหาวัสดุที่จะนำมาสร้างโครงเครื่องคัดแยก

วัสดุที่นำมาสร้างโครงเครื่องคัดแยกดังรูปที่ 3.1 มีดังนี้

1. เหล็กกล่อง ขนาด 1 นิ้ว x 1 นิ้ว ยาว 16 เมตร
2. เหล็กแผ่น ขนาด 1 นิ้ว ยาว 2 เมตร
3. เหล็กฉาก ขนาด 1 นิ้ว ยาว 2 เมตร
4. อลูมิเนียมฉาก ขนาด 1 นิ้ว ยาว 7 เมตร

3.2.2 การออกแบบชุดควบคุม

1. ออกแบบชุดควบคุมมอเตอร์ของชุดสายพานและชุดผลัก

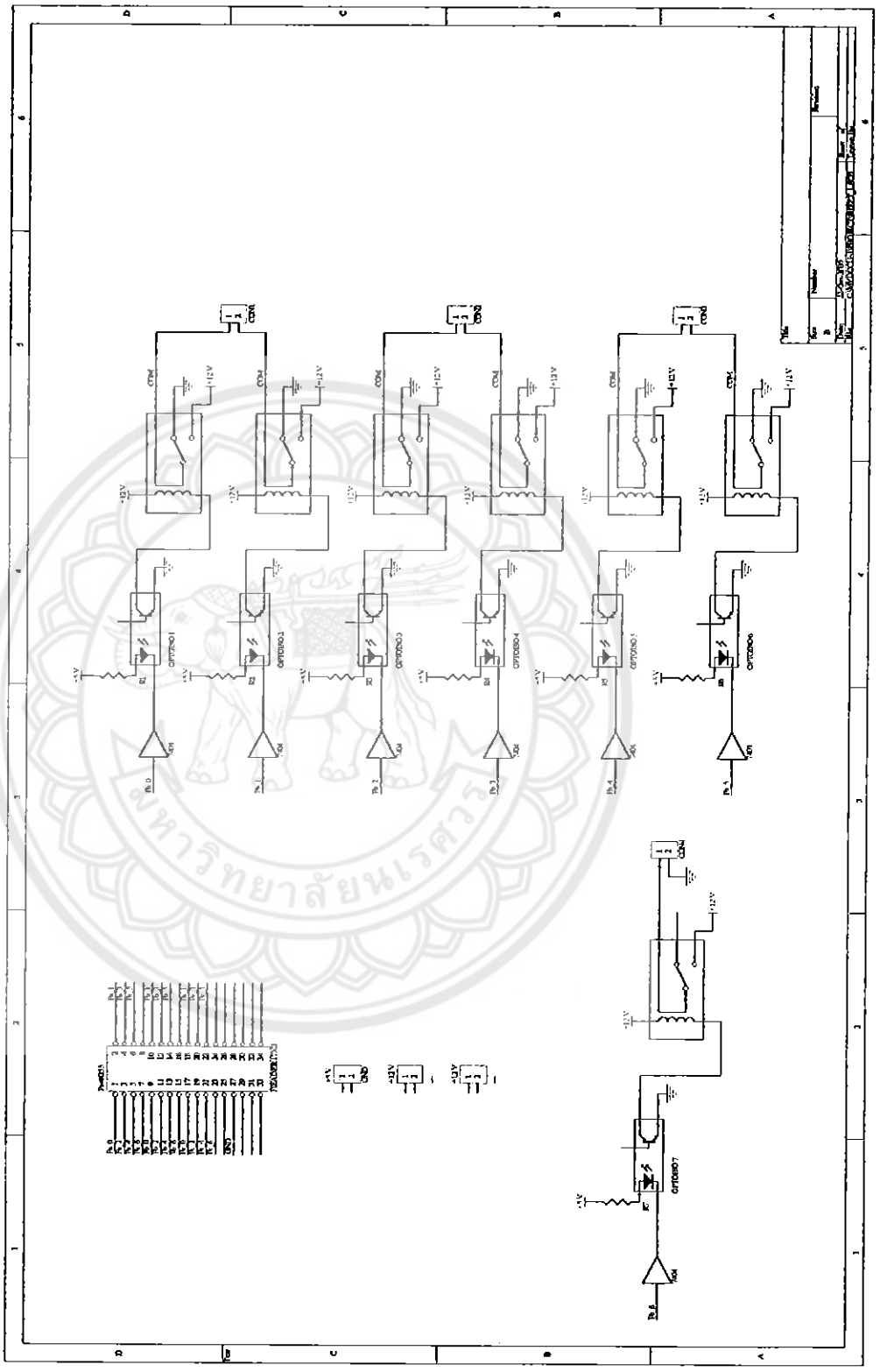
คำอธิบายการทำงานของชุดควบคุมมอเตอร์ของชุดสายพานและชุดผลัก

วงจรควบคุมมอเตอร์ ควบคุมด้วยพอร์ตของไมโครคอนโทรเลอร์ โดยใช้พอร์ต Pb เป็นตัวควบคุม โดยชุดผลักซึ่งมี 3 ชุด จะใช้ 2 พอร์ตในการควบคุมชุดผลักหนึ่งชุด รวมเป็น 6 พอร์ตและในส่วนของมอเตอร์ควบคุมสายพาน เนื่องจากสายพานหมุนเพียงทางเดียวจึงใช้เพียงพอร์ตเดียวในการควบคุม

วงจรนี้ได้ใช้ relay เป็นตัวควบคุมโดยมอเตอร์ของชุดผลักนั้นจะหมุนได้ 2 ทิศทาง ตามเข็มและทวนเข็ม ดังนั้นจึงใช้ 2 พอร์ตควบคุม โดยในชุดผลักชุดที่ 1 จะใช้พอร์ต Pb_0 และ Pb_1 ควบคุมชุดผลักชุดที่ 2 จะใช้พอร์ต Pb_2 และ Pb_3 ควบคุม ชุดผลักชุดที่ 3 จะใช้พอร์ต Pb_4 และ Pb_5 ควบคุม ในส่วนของมอเตอร์ควบคุมสายพานจะใช้พอร์ต Pb_6 เริ่มต้นในสถานนะของ Pb = 0 การทำงานของชุดผลักเราจะสั่งให้ Pb_0 = 1, Pb_1 = 0 เมื่อต้องการให้ชุดผลักที่ 1 ผลักวัตถุ, Pb_2 = 1, Pb_3 = 0 เมื่อต้องการให้ชุดผลักที่ 2 ผลักวัตถุ, Pb_4 = 1, Pb_5 = 0 เมื่อต้องการให้ชุดผลักที่ 3 ผลักวัตถุ ในส่วนของการหยุดชุดผลักนั้นชุดผลักที่ 1 จะสั่งโดยให้ Pb_0 = 0, Pb_1 = 1 ชุดผลักที่ 2 จะสั่งโดยให้ Pb_2 = 0, Pb_3 = 1 ชุดผลักที่ 3 จะสั่งโดยให้ Pb_4 = 0, Pb_5 = 1 สนส่วนของมอเตอร์ควบคุมสายพานนั้นจะควบคุมด้วยพอร์ต Pb_6 โดยให้ Pb_6 = 1 เมื่อต้องการให้หมุนไปข้างหน้า และให้ Pb_6 = 0 เมื่อต้องการหยุด

4K00199
S
675
ว 475
2543

ห้องสมุดคณะวิศวกรรมศาสตร์



รูปที่ 3.2 ภาพแสดงวงจรชุดควบคุมมอเตอร์ของชุดสายพานและชุดผลัก

2. ออกแบบระบบตรวจสอบวัตถุ (sensor)

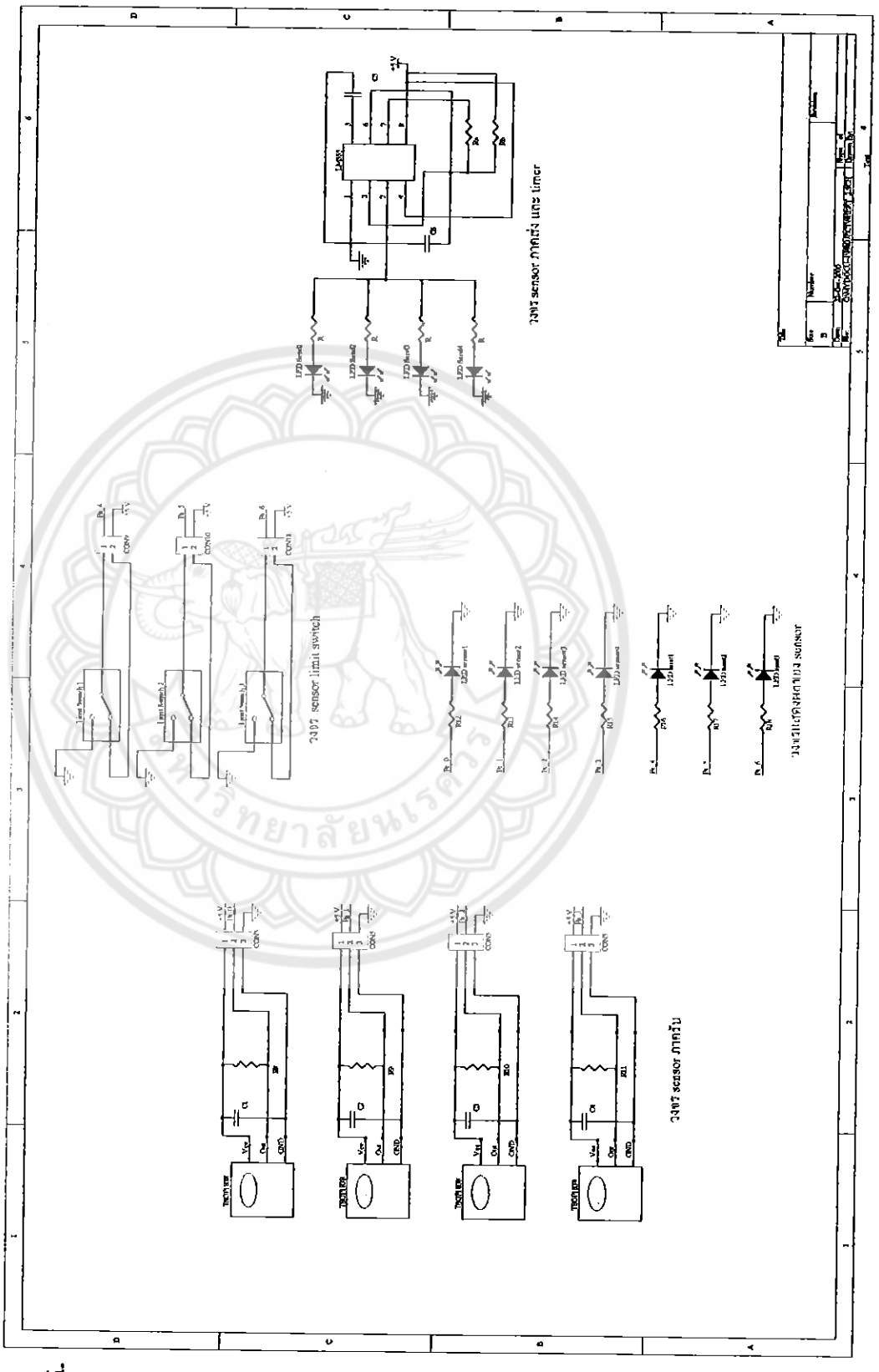
คำอธิบายการทำงานของระบบตรวจสอบวัตถุ

วงจรของระบบตรวจสอบวัตถุทำงานโดยภาคส่งนั้นจะมีการต่อเข้ากับชุดควบคุมความถี่เพื่อให้อัตราการส่งแสงเป็นไปตามความถี่ที่ภาครับต้องการ โดยภาคส่งนั้นจะเปล่งแสงออกมาตลอด ต่อเมื่อมีวัตถุเข้ามาทำให้เกิดการสะท้อนแสงเข้าสู่ภาครับนั้น ภาครับจึงจะทำงาน สถานะที่ทำงานคือ logic 1 และส่งค่า output เข้าไปทางพอร์ต Pa ของไมโครคอนโทรเลอร์ เพื่อส่งค่าไปให้โปรแกรมประมวลผลทราบว่ามีวัตถุมาอยู่ ณ. ตำแหน่งที่ให้ตรวจสอบแล้ว ในขณะเดียวกันไมโครคอนโทรเลอร์ก็จะส่งค่าออกมาที่พอร์ต Pc เพื่อแสดงให้เห็นว่าได้รับค่าไปทำงานแล้ว ซึ่งค่าที่ส่งออกมาทางพอร์ต Pc นั้น วงจรแสดงผลของ sensor ก็จะรับเป็นค่าอินพุตเข้าไปเพื่อบอกสถานะให้ทราบว่าไมโครคอนโทรเลอร์ ได้รับค่าจากภาครับเข้าไปทำงานแล้ว

3. ออกแบบการติดตั้งชุดตรวจสอบการกลับสู่ตำแหน่งเริ่มต้นของชุดผลัก

คำอธิบายการทำงานของชุดตรวจสอบการกลับสู่ตำแหน่งเริ่มต้นของชุดผลัก

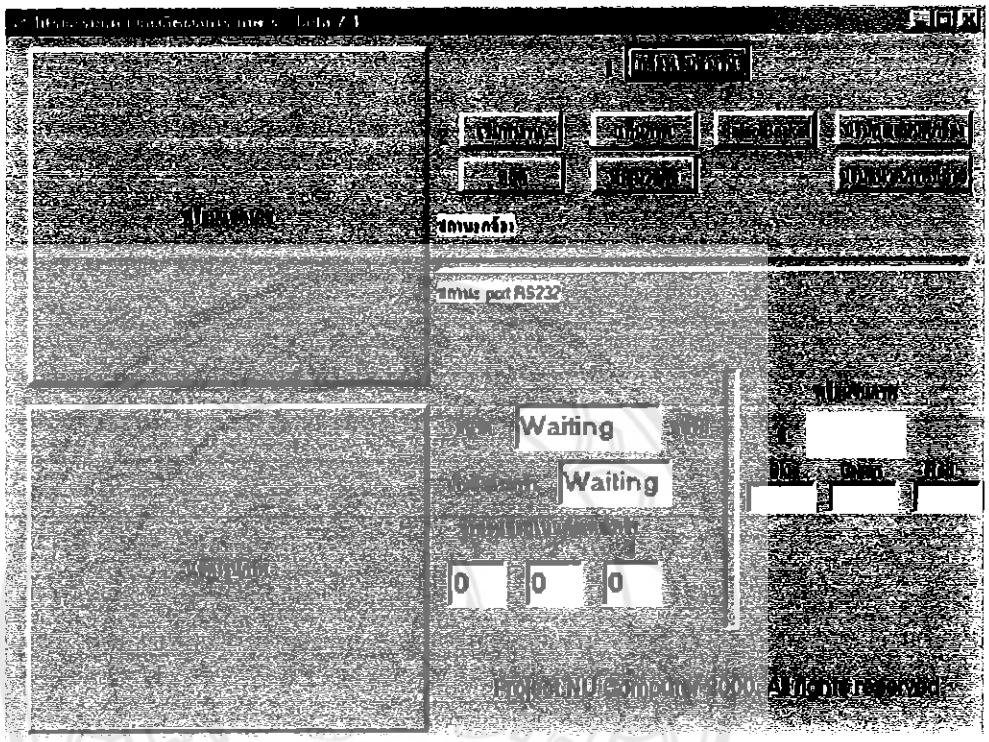
วงจร sensor limit switch นั้นจะทำงานเมื่อแกนผลักของชุดผลักกลับมาสู่ตำแหน่งเริ่มต้นหลังการผลักวัตถุตามคำสั่งคัดแยกก่อนหน้านั้นแล้ว โดยการส่งค่า output เข้าไปที่พอร์ต Pa4 – Pa6 เพื่อไปสั่งให้ตัดการจ่ายไปแก่มอเตอร์จนกว่าจะมีคำสั่งให้ผลักวัตถุสั่งกลับมาใหม่ ซึ่งจะทำให้แกนผลักของชุดผลักหยุดที่ตำแหน่งเริ่มต้นตามต้องการและรอรับคำสั่งการผลักวัตถุต่อไป โดยชุดผลักชุดที่ 1 ส่งค่า output เข้าไปที่พอร์ต Pa4 ชุดผลักชุดที่ 2 ส่งค่า output เข้าไปที่พอร์ต Pa5 และชุดผลักชุดที่ 3 ส่งค่า output เข้าไปที่พอร์ต Pa6



รูปที่ 3.3 ภาพแสดงวงจรของชุด sensor และวงจรของ sensor limit switch

3.2.3 การออกแบบส่วนที่ติดต่อกับผู้ใช้

1. ออกแบบหน้าตาของตัวที่จะติดต่อกับผู้ใช้งานผ่านทางคอมพิวเตอร์



รูปที่ 3.4 ภาพแสดงส่วนของการแสดงผลและติดต่อกับผู้ใช้

2. ออกแบบแนวคิดการทำงานของโปรแกรมที่จะประมวลผล

แนวคิดที่นำมาใช้ในการเขียนโปรแกรมประมวลผลคือ การตรวจสอบเส้นขอบของภาพที่นำมาวิเคราะห์โดยการเปรียบเทียบสีของตัวภาพกับฉาก พอได้ขอบเขตของภาพมาแล้วก็ทำการนับจำนวน pixel ของภาพ ซึ่งในโครงงานนี้ภาพที่นำมาวิเคราะห์คือภาพของมะม่วงที่กล้องจับภาพไว้และนำมาวิเคราะห์เพื่อระบุขนาด แล้วจึงส่งเป็นคำสั่งไปบอกเครื่องคัดแยกว่าเป็นมะม่วงขนาดไหน

3.3 การจัดสร้างชิ้นงาน

การจัดสร้างดำเนินการแบ่งเป็น 3 ส่วน คือ ส่วนของตัวโครงเครื่องคัดแยก ส่วนของชุดควบคุมต่างๆ ส่วนของโปรแกรมและการประมวลผล ซึ่งดำเนินการสร้างพร้อมกันโดยแบ่งตามช่วงเวลา ดังนี้

3.3.1 ช่วงเดือน มีนาคม - เมษายน

3.3.1.1 ส่วนของตัวโครงเครื่องคัดแยก

จัดสร้างโครงเครื่องคัดแยกตามแบบที่วางไว้จนเป็นโครงเครื่องคัดแยกที่เสร็จสมบูรณ์ และทำการจัดทำชุดผลักวัตถุสามชุดตามที่ออกแบบไว้จนเสร็จ

3.3.1.2 ส่วนของชุดควบคุม

จัดทำชุดควบคุมมอเตอร์ของชุดสายพาน และชุดควบคุมมอเตอร์ของชุดคัดแยกวัตถุ

3.3.1.3 ส่วนของโปรแกรมและการประมวลผล

ทดลองเขียนโปรแกรมตามหลักแนวคิดที่ศึกษาไว้ และศึกษาหลักการที่นำมาใช้ให้ชัดเจนยิ่งขึ้น

3.3.2 ช่วงเดือน พฤษภาคม – มิถุนายน

3.3.2.1 ส่วนของตัวโครงเครื่องคัดแยก

ติดตั้งชุดผลัก, มอเตอร์ควบคุมชุดผลัก, ติดตั้งสายพาน, และมอเตอร์ควบคุมชุดสายพาน

3.3.2.2 ส่วนของชุดควบคุม

ทดลองการทำงานของชุดควบคุมการทำงานของมอเตอร์ที่ขับเคลื่อนสายพานและขับเคลื่อนชุดผลัก และจัดทำชุดควบคุมตัวตรวจจับวัตถุ(sensor)ที่ทำงานแบบสะท้อน

3.3.2.3 ส่วนของโปรแกรมและการประมวลผล

เขียนโปรแกรมที่ใช้วิเคราะห์ภาพโดยยังไม่มีส่วนส่งค่าถึงเครื่องคัดแยก และทดลองการทำงานของโปรแกรมส่วนที่ใช้วิเคราะห์ภาพเพื่อแก้ไขและปรับปรุงข้อผิดพลาด ก่อนการนำมาเชื่อมต่อเพื่อส่งข้อมูลให้แก่เครื่องคัดแยก

3.3.3 ช่วงเดือน กรกฎาคม - สิงหาคม

3.3.3.1 ส่วนของตัวโครงเครื่องคัดแยก

ติดตั้งชุดตรวจจับวัตถุแบบสะท้อน กับชุดตรวจสอบการกลับเข้าสู่ตำแหน่งเริ่มต้นของชุดผลัก และทำการทดสอบชุดควบคุมทั้งสอง

3.3.3.2 ส่วนของชุดควบคุม

ทดลองการทำงานของชุดตรวจจับวัตถุแบบสะท้อน และชุดตรวจสอบการกลับเข้าสู่ตำแหน่งเริ่มต้นของชุดผลัก พร้อมกับการทดลองการทำงานของชุดส่งข้อมูลแบบRS232 เพื่อเตรียมติดตั้ง

3.3.3.3 ส่วนของโปรแกรมและการประมวลผล

เขียนโปรแกรมส่วนที่ทำการส่งค่าจากผลการวิเคราะห์ภาพของโปรแกรมส่วนที่วิเคราะห์ภาพ เพื่อส่งไปสั่งงานเครื่องคัดแยก และทดลองโปรแกรมทั้งสองส่วนให้ทำงานร่วมกันอย่างราบรื่น

3.3.4 ช่วงเดือนกันยายน

3.3.4.1 ส่วนของตัวโครงเครื่องคัดแยก

ตรวจสอบความเรียบร้อยพร้อมทาสีเพื่อเตรียมส่งชิ้นงาน และทดลองการทำงานพร้อมกับส่วนชุดควบคุมกับส่วนโปรแกรมและการประมวลผล เพื่อเก็บค่าผลการทดลองส่งพร้อมการส่งชิ้นงาน

3.3.4.2 ส่วนของชุดควบคุม

ตรวจสอบดูแลแก้ไขข้อผิดพลาดขณะทดลองขั้นสุดท้ายพร้อมกับส่วนเครื่องคัดแยกกับส่วนโปรแกรมและการประมวลผล พร้อมเก็บค่าผลการทดลอง

3.3.4.3 ส่วนของ โปรแกรมและการประมวลผล

แก้ไขและปรับปรุงข้อผิดพลาดขณะทำการทดลองพร้อมกับเครื่องคัดแยกและชุดควบคุมต่างๆ พร้อมเก็บผลการทดลอง เพื่อแก้ไขให้พร้อมในการส่งชิ้นงาน

3.4 ขั้นตอนการทดลอง

การทดลองดำเนินการเป็นขั้นตอนดังต่อไปนี้

1. นำมะม่วงมา 2 พันธุ์ พันธุ์ละ 10 ผล
2. นำมะม่วงพันธุ์แรกมาทดลองก่อน โดยเริ่มจากการวางมะม่วงตรงสายพานลำเลียงทีละผล
3. จากนั้นรอกล้องจับภาพ และบันทึกผลว่าโปรแกรมประมวลผลให้อยู่ในขนาดใด และเครื่องคัดแยกทำการการคัดแยกได้ตรงกับการประมวลผลหรือไม่
4. นำมะม่วงพันธุ์แรกผลต่อไปวางตรงสายพานลำเลียงและบันทึกผลจนครบทั้ง 10 ผล
5. ทำการทดลองตามขั้นที่2-ขั้นที่4 อีก 2 ครั้ง
6. เปลี่ยนมะม่วงเป็นพันธุ์ที่สอง และทำการทดลองตามขั้นที่ 2 –ขั้นที่ 5

บทที่ 4

ผลการทดลองและผลการวิเคราะห์

4.1 ผลการทดลอง

ตารางที่ 4.1 แสดงผลการทดลองของมะม่วงพันธุ์แรก

	ครั้งที่ 1(ผล)		ครั้งที่ 2(ผล)		ครั้งที่ 3(ผล)	
	โปรแกรม	เครื่องคัดแยก	โปรแกรม	เครื่องคัดแยก	โปรแกรม	เครื่องคัดแยก
ตระกร้า 1	3	3	4	4	3	3
ตระกร้า 2	3	3	4	4	3	3
ตระกร้า 3	3	3	4	4	3	3

ตารางที่ 4.2 แสดงผลการทดลองของมะม่วงพันธุ์ที่สอง

	ครั้งที่ 1(ผล)		ครั้งที่ 2(ผล)		ครั้งที่ 3(ผล)	
	โปรแกรม	เครื่องคัดแยก	โปรแกรม	เครื่องคัดแยก	โปรแกรม	เครื่องคัดแยก
ตระกร้า 1	5	5	2	2	3	3
ตระกร้า 2	5	5	2	2	3	3
ตระกร้า 3	5	5	2	2	3	3

4.2 วิเคราะห์ผลการทดลอง

จากทั้งผลการทดลองที่ได้ของผลมะม่วงนั้นเมื่อนำผลมะม่วงที่ถูกคัดแยกแล้วทั้งสองพันธุ์มาเปรียบเทียบกันตระกร้าต่อตระกร้าจะพบว่ายังมีข้อผิดพลาดซึ่งเกิดขึ้นจากการมิได้วิเคราะห์วัตถุในแบบ 3 มิติ หากวิเคราะห์จากผลการทดลองที่ได้ ในกรณีของผลมะม่วงนั้นมีข้อผิดพลาดเกิดขึ้นมากในด้านความแตกต่างของปริมาณน้ำหนักของผลมะม่วงแต่ละพันธุ์ในตระกร้าเดียวกันของมะม่วงทั้งสองพันธุ์ ซึ่งควรจะมีปริมาณใกล้เคียงกันในแต่ละผลในตระกร้าเดียวกัน ซึ่งข้อผิดพลาดนี้เกิดขึ้นเนื่องจากผลมะม่วงนั้นเป็นผลไม้ที่มีรูปร่างไม่ค่อยสมมาตร บางผลสั้นแบน บางผลยาวแบน หรือบางผลสั้นหนา บางผลสั้นแบน นอกจากนี้ลักษณะผลของมะม่วงที่ต่างพันธุ์กันก็มีรูปร่างต่างกันตัวอย่างเช่นพันธุ์แรกที่นำมาทดลองผลมีรูปร่างยาวแต่ไม่หนามากในแต่ละผล ในขณะที่พันธุ์ที่สองนั้นผลมีรูปร่างสั้นแต่มีความหนามาก เมื่อเราใช้กล้องเพียงหนึ่งตัวมาจับภาพมาวิเคราะห์และภาพที่ได้เป็นภาพ 2 มิตินั้น ทำให้การคัดแยกถูกทำขึ้นโดยประมวลผลในแบบ 2 มิติ จึงเกิดความผิดพลาดเนื่องจากเราไม่ทราบความหนาของผลมะม่วงด้วย แต่การทราบความหนานั้นต้องใช้การประมวลผลในแบบ 3 มิติ ซึ่งในขอบเขตของโครงงานนี้เราทำการวิเคราะห์ภาพเพียง 2 มิติเท่านั้น แต่ข้อผิดพลาดนี้ก็จะพบน้อยลงถ้าเราทำการคัดแยกมะม่วงพันธ์เดียวกัน

บทที่ 5
บทสรุป

5.1 สรุปผลการทดลอง

จากผลการทดลอง สรุปได้ว่าผลการคัดแยกจากการวิเคราะห์ของโปรแกรมกับการทำงานของเครื่องคัดแยกนั้นมีการสัมพันธ์แม่นยำตรงกันดี และหากนำเครื่องคัดแยกนี้ไปคัดแยกผลิตผลทางการเกษตรที่อยู่ในพันธุ์เดียวกันก็จะลดข้อผิดพลาดด้านปริมาณน้ำหนักได้มากกว่าการคัดแยกผลผลิตทางการเกษตรที่ต่างพันธุ์กัน

5.2 ข้อเสนอแนะ

1. จากผลการทดลองที่ได้เราจะพบว่า เรายังขาดผลในแบบ 3 มิติ ซึ่งนอกเหนือจากขอบเขตโครงงานของกลุ่มผู้จัดทำ หากเรามีการติดตั้ง sensor เข้าไปเพื่อตรวจจับขนาดของผลผลิตทางการเกษตรนั้นๆ และนำผลจากการวิเคราะห์จากภาพที่ถ่ายด้วยกล้องวีดีโอพีซีมารวมกัน จะทำให้เราสามารถแยกผลผลิตได้ตามต้องการมากขึ้น คือได้ทั้งขนาด ความสุกดิบ และรวมทั้งแยกผลที่มีจุดด่างดำที่เป็นจุดเสียของตัวผลผลิตได้อีกด้วย
2. จากข้อเสนอที่ เราอาจไม่ใช้การติดตั้ง sensor แต่อาจเปลี่ยนเป็นการติดตั้งกล้องจำนวนมากขึ้นเพื่อจับภาพในหลายมุม ทว่าวิธีนี้ย่อมเสียค่าใช้จ่ายมากกว่าซึ่งในการนำไปใช้งานอาจต้องคิดถึงความเหมาะสมในด้านต่างๆประกอบด้วยอีกมากมาย
3. นอกจากข้อเสนอแนะทั้งสองข้อแล้ว อาจมีการติดตั้งตัวพลิกผลิตผลเพื่อให้กล้องจับภาพให้ทั่วถึงเพื่อแยกผลที่มีจุดตำหนิออกไปได้แน่นอน ก่อนจะผ่านไปให้ sensor ตรวจจับขนาด ซึ่งวิธีนี้จะให้ผลได้ในขอบเขตที่กว้าง แต่ก็เป็นวิธีที่มีความซับซ้อนและยุ่งยากมากอีกทั้งในการทำงานจริงยังต้องการความรวดเร็วและความเหมาะสมต่อการคัดแยกผลิตผลนั้นๆต่างกันออกไปอีกด้วยจึงทำให้ต้องมีสิ่งที่นำมาพิจารณาร่วมหลายอย่าง

5.3 ปัญหาที่เกิดขึ้นในระหว่างการจัดทำโครงงานและการแก้ไข

1. ปัญหาเกี่ยวกับระบบตรวจจับวัตถุ

   ปัญหา :เนื่องจากแสงอินฟาเรดที่ส่งจากตัวส่งนั้นมีความจ้ามากทำให้ตัวรับสามารถรับแสงได้โดยไม่ต้องรอการสะท้อนจากวัตถุทำให้เครื่องคัดแยกไม่สามารถทำงานตามเป้าหมายได้

   การแก้ปัญหา:ต่อตัวต้านทานแบบปรับค่าได้เข้าไปเพื่อหาค่าที่เหมาะสมในการทำให้เกิดการลดกระแส จากการทดลองปรับค่า ค่าที่เหมาะสมคือค่าประมาณ 200 โอห์ม

2. ปัญหาเกี่ยวกับชุดควบคุมมอเตอร์ของชุดผลัก

   ปัญหา :เนื่องจากชุดควบคุมมอเตอร์ของชุดผลักนั้นทำงานแบบหมุน 2 ทิศทางแต่เนื่องจาก มอเตอร์ที่นำมาใช้นั้นตัวถังของมอเตอร์ได้ทำหน้าที่เป็นกราวด์อยู่แล้วจึงทำให้เกิดการชนกันของกระแสไฟบวกเมื่อชุดควบคุมสั่งให้กลับขั้วเพื่อดึงให้ชุดผลักกลับสู่สถานะเริ่มต้น ทำให้เกิดสายไฟไหม้เสียหาย

   การแก้ปัญหา:ให้ชุดควบคุมบังคับมอเตอร์หมุนในทิศทางเดียวเท่านั้น และเมื่อชุดผลักกลับมาชนลิมิตสวิทย์ที่ติดไว้ก็ให้ชุดควบคุมสั่งติดไฟเพื่อให้ชุดผลักหยุดอยู่ที่ตำแหน่งเริ่มต้นเพื่อรอรับคำสั่งการทำงานต่อไป

บรรณานุกรม

1. Adrian Low INTRODUCTORY COMPUTER VISION AND IMAGE PROCESSING

London : McGRAW-HILL BOOK COMPANY

2. Randy Crane A SIMPLIFIED APPROACH TO IMAGE PROCESSING CLASSICAL AND MODERN TECHNIQUES IN C New Jersey : Hewlett-Packard Company

3. นุกูล กระจาย การเขียนโปรแกรมและประมวลผลข้อมูลด้วยเทอร์โบปาสคาล กรุงเทพฯ : บริษัท ซีเอ็ดยูเคชั่น จำกัด (มหาชน)

4. จักรพงษ์ สุขประเสริฐ, กนกพร ภาวศุทธิกุล, และสัจจะ จรัสรุ่งรวีวร คู่มือการเขียนโปรแกรมด้วย Delphi 4.0 ฉบับสมบูรณ์ กรุงเทพฯ : สำนักพิมพ์อินโฟเพรส

5. สัจจะ จรัสรุ่งรวีวร และจักรพงษ์ สุขประเสริฐ เริ่มต้นอย่างมืออาชีพด้วย Delphi 5 ฉบับสมบูรณ์ กรุงเทพฯ : สำนักพิมพ์อินโฟเพรส

ภาคผนวก

ส่วนของโปรแกรมคอนโทรเลอร์

ส่วนของโปรแกรมวิเคราะห์ภาพ

Datasheet เกี่ยวกับเรื่อง Timer 555

Datasheet เกี่ยวกับเรื่อง TSOP18..

โปรแกรมส่วนคอนโทรเลอร์

```
#include <reg51.h>
#include <absacc.h>
#include <math.h>
#include <string.h>
#include <stdlib.h>
#include <stdio.h>

#define Pa8255     XBYTE[0xE0E0]
#define Pb8255     XBYTE[0xE0E1]
#define Pc8255     XBYTE[0xE0E2]
#define Pcon8255   XBYTE[0xE0E3]

#define LCD_CONTROL   XBYTE[0xE0C0]
#define LCD_BUSY_FLAG XBYTE[0xE0C1]
#define LCD_WRITEDATA XBYTE[0xE0C2]
#define LCD_READDATA  XBYTE[0xE0C3]

xdata char zz[16],j;
data unsigned char clock0;
data unsigned char clock1;

xdata char float_string[16];

void delay(unsigned int time)
{
```

```
data unsigned int i;
for(i= 0;i < time;i++)
        {
        }
}

void wait_for_lcd_ready (void)
        {
while ((LCD_BUSY_FLAG & 0x80) != 0)
        {
        }
//wait util buy flag = 0 (lcd ready)
}

void specified_lcd(void)
        {
//for lcd more than 2 line
//and 5x7 dot per cursor font
LCD_CONTROL=0x38;                    //function set bit5=1
wait_for_lcd_ready ();
}

void clear_lcd(void)
        {
  LCD_CONTROL=0x01;
    //bit0=1 clear lcd
  //and cursor goto top left
wait_for_lcd_ready ();
}
```

```
void cursor_on(void)
{
LCD_CONTROL = 0x0f;                    //bit2=1 lcd monitor on
wait_for_lcd_ready ();
}

void cursor_off(void)
{
LCD_CONTROL = 0x0c;
wait_for_lcd_ready ();
}

void entry_mode_set(void)
{
LCD_CONTROL = 0x02;
//bit0=0 after put data shift cursor to right
//bit1=1 after read/write dd ram address increase1

wait_for_lcd_ready ();
}


void init_lcd(void)
{
specified_lcd();
cursor_on();
clear_lcd();
entry_mode_set();
```

```
}

void WChar(char ch)
{
LCD_WRITEDATA = ch;
wait_for_lcd_ready();
}

void gotoxy(unsigned char row, unsigned char col)
{
if (row == 1) LCD_CONTROL = 0x80+col-1; else
if (row == 2) LCD_CONTROL = 0xc0+col-1; else
if (row == 3) LCD_CONTROL = 0x90+col-1; else
if (row == 4) LCD_CONTROL = 0xd0+col-1;
wait_for_lcd_ready ();
}

void WCharxy(unsigned char row,unsigned char col,unsigned char ch)
{
gotoxy(row,col);
WChar(ch);
}

void string_to_lcd (unsigned char row,unsigned char col,char *st)
{
data unsigned char mcol,colfinish,len;

len = strlen(st);
```

```
colfinish=col+len-1;
if (colfinish > 16)  colfinish=16;

for (mcol=col; mcol<=colfinish; mcol++)
        {
    WCharxy(row,mcol,st[mcol-col]);
        }
}

void string_to_lcd32 (char *st)
{
data unsigned char mcol,colfinish,len,i;
    xdata char display_lcd[32];

    for (i=0;i<=31;i++)
        {
        display_lcd[i]=' ';
        }
    len = strlen(st);

    if (len>32) len=32;
for (i=0;i<=len-1;i++)
        {
    display_lcd[i]=st[i];
        }

for (mcol=1; mcol<=16; mcol++)
        {
 WCharxy(1,mcol,display_lcd[mcol-1]);
        }
```

```
for (mcol=1; mcol<=16; mcol++)
{
WCharxy(2,mcol,display_lcd[(mcol+16)-1]);
}
}

void string_to_lcd32RotateLeft (char *st)
{
data unsigned char mcol,colfinish,len,i,temp;
xdata char display_lcd[32];

for (i=0;i<=31;i++)
{
display_lcd[i]=' ';
}
len = strlen(st);

if (len>32) len=32;
for (i=0;i<=len-1;i++)
{
display_lcd[i]=st[i];
}

while (1)
{
string_to_lcd32(display_lcd);

temp=display_lcd[0];
for (i=1;i<=31;i++)
```

```
{
display_lcd[i-1]=display_lcd[i];
}
display_lcd[31]=temp;

delay(10000);
}

}

char getkey()
{
unsigned char c;
while(!RI)
{
}
c = SBUF;
RI = 0;
return(c);
}

void saipan_run(void)
{
Pb8255=0x40; //forward
}

void saipan_stop(void)
{
Pb8255=0x00; //break all
}
```

```
void push1(void) //saipan running
{
Pb8255=0x42;      //push forward
delay(30000);
while(1)
{
if((Pa8255&0x10)==0)
{
saipan_run();
break;
}

}
}
void push2(void) //saipan running
{
Pb8255=0x48;
delay(30000);
while(1)
{
if((Pa8255&0x20)==0)
{
saipan_run();
break;
}

}
}
```

```
void push3(void) //saipan running
{
Pb8255=0x60;
delay(30000);
while(1)
{
if((Pa8255&0x40)==0)
{
saipan_run();
break;
}
}

}

void init_start(void)
{
delay(30000);
push1();      //saipan start run
delay(30000);
push2();
delay(30000);
push3();
delay(30000);
string_to_lcd(1,1,"Ready to run");
}

void init_serial(void)
{
```

```
SCON = 0x52;
PCON = 0x00;
TMOD = 0x20;
TH1 = 0xfd;
TR1 = 1;
//  TI = 1;
//  RI = 0;
}

void main(void)
{
xdata unsigned char i;
xdata unsigned int j;

delay(20000);
init_lcd();
Pcon8255=0x80;
init_serial();
string_to_lcd(2,1,"on on on");
Pa8255=0xff;
Pb8255=0x00;
Pc8255=0x55;
delay(10000);
saipan_run();
init_start(); //saipan running
saipan_stop();


while(1)
{
```

```
i=getkey();
j=i;
switch (i)
{
case'0':
{
saipan_run();
while(1)
{
if((Pa8255&0x01)==0)
{
saipan_stop();
delay(10000);
Pc8255=0x71;
printf("take");
string_to_lcd(1,1,"camera take");
break;
}

}break;
break;
}

case'1':
{
saipan_run();
while(1)
{
if((Pa8255&0x02)==0)
{
```

```
//          saipan_stop();

            delay(30000);
            Pc8255=0x72;
            push1();
            printf("push1");
        string_to_lcd(1,1,"push1 finish");
            break;
            }
            }break;
            break;
            }

            case'2':
            {
        saipan_run();
            while(1)
            {
        if((Pa8255&0x04)==0)
            {
//          saipan_stop();

            delay(30000);
            Pc8255=0x74;
            push2();
            printf("push2");
        string_to_lcd(1,1,"push2 finish");
            break;
            }
            }break;
```

```
                break;
                }

                case'3':
                {
            saipan_run();
                while(1)
                {
            if((Pa8255&0x08)==0)
                {
        //          saipan_stop();

                delay(30000);
                Pc8255=0x78;
                push3();
                printf("push3");
        string_to_lcd(1,1,"push3 finish");
                break;
                }
                }break;
                break;
                }
                case'4':
                {
            saipan_stop();
                break;
                }break;

                }//end case
```

```
} //end while


} //end main
```

โปรแกรมส่วนวิเคราะห์ภาพ

```
unit camera1;

interface

uses
 Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
 StdCtrls, ExtCtrls,VideoCap, ComCtrls, Async32;

type
 TForm1 = class(TForm)
  Label1: TLabel;
  Image3: TImage;
  Label2: TLabel;
  Label3: TLabel;
  Label4: TLabel;
  Label6: TLabel;
  Panel1: TPanel;
  videoarea: TPanel;
  Image1: TImage;
  Button1: TButton;
  Button2: TButton;
  Button3: TButton;
  Button4: TButton;
  Button5: TButton;
  Button6: TButton;
  ProgressBar1: TProgressBar;
  Edit1: TEdit;
  EditR: TEdit;
  Button7: TButton;
```

```
Comm1: TComm;
EditSize: TEdit;
Label24: TLabel;
Label27: TLabel;
EditG: TEdit;
EditB: TEdit;
ShapeShowColor: TShape;
Label25: TLabel;
Label26: TLabel;
Label28: TLabel;
EditGain: TEdit;
Panel3: TPanel;
Label29: TLabel;
Button9: TButton;
Bas1: TEdit;
SelectBasket: TButton;
Bas2: TEdit;
Bas3: TEdit;
Label7: TLabel;
ProgressBar2: TProgressBar;
Label5: TLabel;
Label8: TLabel;
Label9: TLabel;
Label10: TLabel;
Label11: TLabel;
Label12: TLabel;
ColorDialog1: TColorDialog;
procedure Button1Click(Sender: TObject);
procedure CAPStatus(Sender: TObject);
procedure Button2Click(Sender: TObject);
```

```
    procedure Button3Click(Sender: TObject);
    procedure Button4Click(Sender: TObject);
    procedure Button5Click(Sender: TObject);
    procedure Button6Click(Sender: TObject);
    procedure Image3MouseUp(Sender: TObject; Button: TMouseButton;
      Shift: TShiftState; X, Y: Integer);
    procedure Button7Click(Sender: TObject);
    procedure Button8Click(Sender: TObject);
    procedure Comm1RxChar(Sender: TObject; Count: Integer);
    procedure FormCreate(Sender: TObject);
    procedure EditGainKeyDown(Sender: TObject; var Key: Word;
      Shift: TShiftState);
    procedure ShapeShowColorMouseDown(Sender: TObject;
      Button: TMouseButton; Shift: TShiftState; X, Y: Integer);
    procedure Button9Click(Sender: TObject);
    procedure SelectBasketClick(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }


  end;
var
  Form1: TForm1;
  iSelcolor,b1,b2,b3 : Integer;
  i,j,iCount,iColor,iBlckHight,iBlckWidth : Integer;
  gain:byte;
  Wsize : longint;
```

```
implementation

{$R *.DFM}



procedure TForm1.CAPStatus(Sender: TObject);
begin
        Panel1.Color := clBtnFace;
        Panel1.Refresh;
end;

procedure TForm1.Button1Click(Sender: TObject);
var
 MyCapStatusProc : TCapStatusProc;


// E: Exception;
begin
                // Start CAP - Video
        CapSetVideoArea( VideoArea );
 CapSetInfoLabel( Label1 );
 MyCapStatusProc := CAPStatus;
        CapSetStatusProcedure( MyCapStatusProc );
        if CapOpenDriver then
 begin
                CapSetCapSec( 15 * 3 );
                CapShow;
        end;
```

```
  Comm1.DeviceName := 'Com1';
  try
    Comm1.Open;
    Label27.Caption := 'Device ready: ' + Comm1.DeviceName;
    Label27.Caption := GetProviderSubtypeName(Comm1.ProviderSubtype) + ' พร้อม';
  except
    on E: ECommError do
      Edit1.Text := E.Message;
  end;
end;

procedure TForm1.Button2Click(Sender: TObject);
var
  S: string;
  Count: Integer;
begin
// S := '4';
// Count := Comm1.Write(S[1], Length(S));
// if Count = -1 then
// Label10.Caption := 'Error open port Com1'
// else Label10.Caption  := 'ส่งไปแล้ว ' + IntToStr(Count) + ' อักษร คือคำสั่งหยุด';

  Comm1.Close;
  Label27.Caption := 'Port RS232 ปิดแล้ว';
  CapCloseDriver;
end;

procedure TForm1.Button3Click(Sender: TObject);
begin
        CapDlgVSource;
```

```
end;

procedure TForm1.Button4Click(Sender: TObject);
begin
        CapDlgVFormat;
end;
procedure TForm1.Button5Click(Sender: TObject);
var
// iCount : Integer;
 SingleImageFileName : string;
begin
//    iHighcolor := iSelcolor + 10000;

  ProgressBar1.Max := Image3.Picture.Height;
  Label24.Visible := False;
                // Save Video as Bitmap to file in TEMP-Path
  SingleImageFileName := 'Image1.bmp';
  CapSetSingleImageFileName( SingleImageFileName );
        CapGrabSingleFrame;
        CapSetVideoLive;
  Image3.Picture.LoadFromFile('Image1.bmp');
  Panel3.Visible := False;
  end;

procedure TForm1.Button6Click(Sender: TObject);
var
   bitmap:tbitmap;
   r,g,b:byte;
   col:dword;
   sumR,sumG,sumB,tmp:double;
```

```
  i,j:integer;
// Wsize:longint;
begin
  bitmap:=tbitmap.Create;
  bitmap.LoadFromFile('Image1.bmp');
  WSize:=0;
  sumR:=0;SumG:=0;SumB:=0;
  for i:=0 to bitmap.Width-1 do
    for j:=0 to bitmap.Height-1 do
    begin
      col:=bitmap.Canvas.Pixels[i,j];


      r:=lo(col);
      g:=hi(col);
      b:=hiword(col);


      sumR:=sumR+r;
      sumG:=sumG+g;
      sumB:=sumB+b;
      r:=((30*r)+(59*g)+(11*b))div 100;
      if r>Gain then
      begin
        inc(Wsize);
        bitmap.Canvas.Pixels[i,j]:=$ffffff;
      end else bitmap.Canvas.Pixels[i,j]:=0;
    end;
  tmp:=(bitmap.Height+1)*(bitmap.Width+1);
// edit2.text:=format('%.0f',[SumR]);
```

```
//  edit3.text:=format('%.0f',[SumG]);
//  edit4.text:=format('%.0f',[SumB]);
   sumR:=sumR/tmp;
   sumG:=sumG/tmp;
   sumB:=sumB/tmp;

   editR.text:=format('%.0f',[SumR]);
   editG.text:=format('%.0f',[SumG]);
   editB.text:=format('%.0f',[SumB]);
   r:=strtoint(editR.text);
   g:=strtoint(editG.text);
   b:=strtoint(editB.text);
   shapeshowcolor.Brush.Color:=rgb(r,g,b);
   editsize.Text:=inttostr(76800 - Wsize);
   image3.Picture.Bitmap:=bitmap;
   bitmap.Free;
   Label10.Caption := 'วิเคราะห์ภาพเสร็จแล้ว';
end;

procedure TForm1.Image3MouseUp(Sender: TObject; Button: TMouseButton;
 Shift: TShiftState; X, Y: Integer);
//var
//  i,j : Integer;
begin
//  Image3.Hint := IntToStr(X) + IntToStr(Y);
//  iSelcolor := Image3.Canvas.Pixels[X,Y];
//  Edit2.Text := IntToStr(iSelcolor);
//  for i := 0 to Image2.Picture.Height do
//  begin
//    for j := 0 to Image2.Picture.Width do
```

```
//    begin
//      Image2.Canvas.Pixels[j,i] := iSelcolor;
//    end;
//  end;
end;

procedure TForm1.Button7Click(Sender: TObject);
var
  S: string;
  Count: Integer;
begin
  S := Edit1.Text;
  Count := Comm1.Write(S[1], Length(S));
  if Count = -1 then
  Label27.Caption := 'Error open port Com1'
  else Label27.Caption  := 'ส่งไปแล้ว ' + IntToStr(Count) + ' อักษร';
end;

procedure TForm1.Button8Click(Sender: TObject);
var
  S: string;
  Count: Integer;
begin
  S := '4';
  Count := Comm1.Write(S[1], Length(S));
  if Count = -1 then
  Label27.Caption := 'Error open port Com1'
  else Label27.Caption  := 'ส่งไปแล้ว ' + IntToStr(Count) + ' อักษร คือคำสั่งหยุด';
end;
```

```
// receive message for controller
procedure TForm1.Comm1RxChar(Sender: TObject; Count: Integer);
var
  Buffer : array[0..10] of Char;
  Bytes : Integer;

begin

  i := i + 1;
  Fillchar(Buffer, Sizeof(Buffer), 0);
  Bytes := Comm1.Read(Buffer, Count);
  if Bytes = -1 then Label27.Caption := 'ไม่สามารถอ่านข้อมูลเข้า';
  Label27.Caption := 'รับเข้ามาแล้ว '+IntToStr(Count) +' อักษร ' + Buffer;

  if Buffer = 'take' then
  begin
    Label10.Caption := '';
    Button5Click(nil);
    Button6Click(nil);
    SelectBasketClick(nil);
  end;

  if Buffer = 'push1' then
  begin
    Button9Click(nil);
    b1:=(b1+1);
    Bas1.text:=inttostr(b1);

  end;
```

```
  if Buffer = 'push2' then
  begin
    Button9Click(nil);
    b2:=(b2+1);
    Bas2.text:=inttostr(b2);
  end;

  if Buffer = 'push3' then
  begin
    Button9Click(nil);
    b3:=(b3+1);
    Bas3.text:=inttostr(b3);
  end;
end;

procedure TForm1.FormCreate(Sender: TObject);
begin

  b1:=0;
  b2:=0;
  b3:=0;
  i := 0;
  gain:=$80;
end;

procedure TForm1.EditGainKeyDown(Sender: TObject; var Key: Word;
  Shift: TShiftState);
begin
  if key=13 then
```

```
  begin
    gain:=strtointdef(editGain.text,$80);
    button6.Click;
  end;
end;

procedure TForm1.ShapeShowColorMouseDown(Sender: TObject;
 Button: TMouseButton; Shift: TShiftState; X, Y: Integer);
begin
  editgain.Visible:=not editgain.Visible;
end;

procedure TForm1.Button9Click(Sender: TObject);
var
 S: string;
 Count: Integer;

begin
  S := '0';
 Count := Comm1.Write(S[1], Length(S));
 Label27.Caption  := 'ส่งไปแล้ว ' + IntToStr(Count) + ' อักษร';
 Label10.Caption := 'กำลังรอ sensor ตรวจจับวัตถุ ณ จุดถ่ายภาพ';
end;

// for select basket
procedure TForm1.SelectBasketClick(Sender: TObject);
var
  Size,Basket1,Basket2,Basket3: longint;
  S : string;
  Count : Integer;
```

```
begin
Basket1 := 30000;
Basket2 := 36000;
Basket3 := 76800;
Size:=(76800-Wsize);

if (Size <= Basket1) then
   begin
    Edit1.Text:='1';
    S := Edit1.Text;
    Count := Comm1.Write(S[1], Length(S));
    if Count = -1 then
    Label27.Caption := 'Error open port Com1'
    else Label27.Caption  := 'ส่งไปแล้ว ' + IntToStr(Count) + ' อักษร';
    Exit;
   end;
if ((Size > Basket1) and (Size <= Basket2)) then
   begin
    Edit1.Text:='2';
    S := Edit1.Text;
    Count := Comm1.Write(S[1], Length(S));
    if Count = -1 then
    Label27.Caption := 'Error open port Com1'
    else Label27.Caption  := 'ส่งไปแล้ว ' + IntToStr(Count) + ' อักษร';
    Exit;
   end;
if ((Size > Basket2) and (Size <= Basket3)) then
   begin
    Edit1.Text:='3';
    S := Edit1.Text;
```

```
    Count := Comm1.Write(S[1], Length(S));
    if Count = -1 then
    Label27.Caption := 'Error open port Com1'
    else Label27.Caption  := 'ส่งไปแล้ว ' + IntToStr(Count) + ' อักษร';
    Exit;
   end;

end;

end. // end main
```

May 1997

National Semiconductor

LM555/LM555C Timer

General Description

The LM555 is a highly stable device for generating accurate time delays or oscillation. Additional terminals are provided for triggering or resetting if desired. In the time delay mode of operation, the time is precisely controlled by one external resistor and capacitor. For astable operation as an oscillator, the free running frequency and duty cycle are accurately controlled with two external resistors and one capacitor. The circuit may be triggered and reset on falling waveforms, and the output circuit can source or sink up to 200 mA or drive TTL circuits.

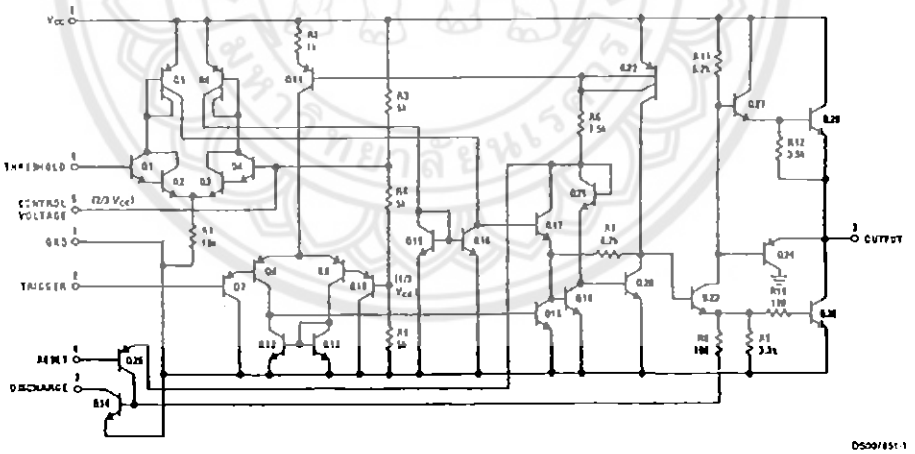
Features

- Direct replacement for SE555/NE555
- Timing from microseconds through hours
- Operates in both astable and monostable modes
- Adjustable duty cycle
- Output can source or sink 200 mA
- Output and supply TTL compatible
- Temperature stability better than 0.005% per °C
- Normally on and normally off output
- Available in 8 pin MSOP package

Applications

- Precision timing
- Pulse generation
- Sequential timing
- Time delay generation
- Pulse width modulation
- Pulse position modulation
- Linear ramp generator

Schematic Diagram



DS007851-1

© 1997 National Semiconductor Corporation DS007851

Absolute Maximum Ratings (Note 2)

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/ Distributors for availability and specifications.

Supply Voltage	+18V
Power Dissipation (Note 3)	
LM555H, LM555CH	760 mW
LM555, LM555CN	1180 mW
LM555CMM	613 mW
Operating Temperature Ranges	
LM555C	0°C to +70°C
LM555	−55°C to + 125°C
Storage Temperature Range	−65°C to +150°C
Soldering Information	
Dual-In-Line Package	
Soldering (10 Seconds)	260°C
Small Outline Packages	
(SOIC and MSOP)	
Vapor Phase (60 Seconds)	215°C
Infrared (15 Seconds)	220°C

See AN-450 "Surface Mounting Methods and Their Effect on Product Reliability" for other methods of soldering surface mount devices.

Electrical Characteristics (Notes 1, 2)

(T_A = 25°C, V_{CC} = +5V to +15V, unless otherwise specified)

Parameter	Conditions	Limits						Units
		LM555			LM555C			
		Min	Typ	Max	Min	Typ	Max	
Supply Voltage		4.5		18	4.5		16	V
Supply Current	V_{CC} = 5V, R_L = ∞		3	5		3	6	mA
	V_{CC} = 15V, R_L = ∞ (Low State) (Note 4)		10	12		10	15	mA
Timing Error, Monostable								
Initial Accuracy			0.5			1		%
Drift with Temperature	R_A = 1k to 100 kΩ, C = 0.1 µF, (Note 5)		30			50		ppm/°C
Accuracy over Temperature			1.5			1.5		%
Drift with Supply			0.05			0.1		%/V
Timing Error, Astable								
Initial Accuracy			1.5			2.25		%
Drift with Temperature	R_A, R_B = 1k to 100 kΩ, C = 0.1 µF, (Note 5)		90			150		ppm/°C
Accuracy over Temperature			2.5			3.0		%
Drift with Supply			0.15			0.30		%/V
Threshold Voltage			0.667			0.667		x V_{CC}
Trigger Voltage	V_{CC} = 15V	4.8	5	5.2		5		V
	V_{CC} = 5V	1.45	1.67	1.9		1.67		V
Trigger Current			0.01	0.5		0.5	0.9	µA
Reset Voltage		0.4	0.5	1	0.4	0.5	1	V
Reset Current			0.1	0.4		0.1	0.4	mA
Threshold Current	(Note 6)		0.1	0.25		0.1	0.25	µA
Control Voltage Level	V_{CC} = 15V	9.6	10	10.4	9	10	11	V
	V_{CC} = 5V	2.9	3.33	3.8	2.6	3.33	4	V
Pin 7 Leakage Output High			1	100		1	100	nA
Pin 7 Sat (Note 7)								
Output Low	V_{CC} = 15V, I_7 = 15 mA		150			180		mV
Output Low	V_{CC} = 4.5V, I_7 = 4.5 mA		70	100		80	200	mV

Electrical Characteristics (Notes 1, 2) (Continued)

(T_A = 25°C, V_{CC} = +5V to +15V, unless otherwise specified)

Parameter	Conditions	Limits						Units
		LM555			LM555C			
		Min	Typ	Max	Min	Typ	Max	
Output Voltage Drop (Low)	V_{CC} = 15V							
	I_{SINK} = 10 mA		0.1	0.15		0.1	0.25	V
	I_{SINK} = 50 mA		0.4	0.5		0.4	0.75	V
	I_{SINK} = 100 mA		2	2.2		2	2.5	V
	I_{SINK} = 200 mA		2.5			2.5		V
	V_{CC} = 5V							
	I_{SINK} = 8 mA		0.1	0.25				V
	I_{SINK} = 5 mA					0.25	0.35	V
Output Voltage Drop (High)	I_{SOURCE} = 200 mA, V_{CC} = 15V		12.5			12.5		V
	I_{SOURCE} = 100 mA, V_{CC} = 15V	13	13.3		12.75	13.3		V
	V_{CC} = 5V	3	3.3		2.75	3.3		V
Rise Time of Output			100			100		ns
Fall Time of Output			100			100		ns

Note 1: All voltages are measured with respect to the ground pin, unless otherwise specified.

Note 2: Absolute Maximum Ratings indicate limits beyond which damage to the device may occur. Operating Ratings indicate conditions for which the device is functional, but do not guarantee specific performance limits. Electrical Characteristics state DC and AC electrical specifications under particular test conditions which guarantee specific performance limits. This assumes that the device is within the Operating Ratings. Specifications are not guaranteed for parameters where no limit is given, however, the typical value is a good indication of device performance.

Note 3: For operating at elevated temperatures the device must be derated above 25°C based on a +150°C maximum junction temperature and a thermal resistance of 164°C/W (TO-5), 106°C/W (DIP), 170°C/W (SO-8), and 204°C/W (MSOP) junction to ambient.

Note 4: Supply current when output high typically 1 mA less at V_{CC} = 5V.

Note 5: Tested at V_{CC} = 5V and V_{CC} = 15V.

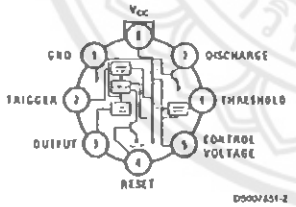
Note 6: This will determine the maximum value of R_A + R_B for 15V operation. The maximum total (R_A + R_B) is 20 MΩ.

Note 7: No protection against excessive pin 7 current is necessary providing the package dissipation rating will not be exceeded.

Note 8: Refer to RETS555X drawing of military LM555H and LM555J versions for specifications.

Connection Diagrams

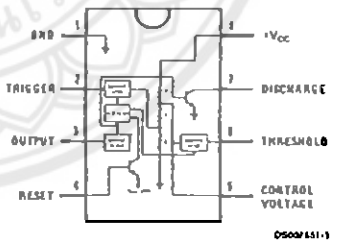
Metal Can Package



D500751-2

Top View
Order Number LM555H or LM555CH
See NS Package Number H08C

Dual-In-Line, Small Outline and Molded Mini Small Outline Packages

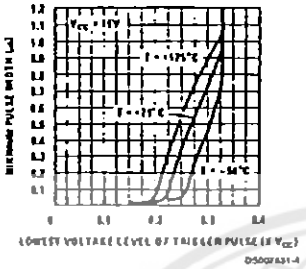


D500751-1

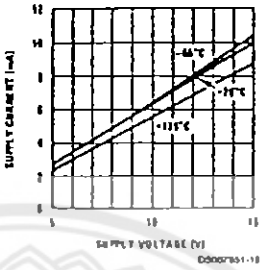
Top View
Order Number LM555J, LM555CJ, LM555CM, LM555CMM or LM555CN
See NS Package Number J08A, M08A, MUA08A or N08E

Typical Performance Characteristics

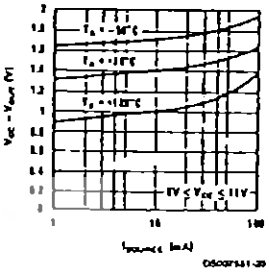
Minimum Pulse Width Required for Triggering



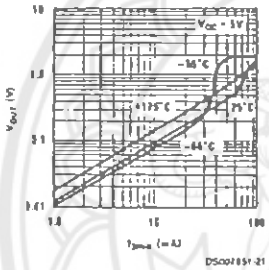
Supply Current vs Supply Voltage



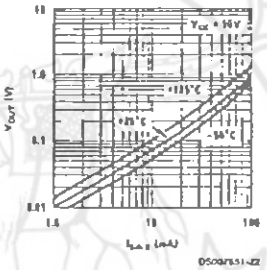
High Output Voltage vs Output Source Current



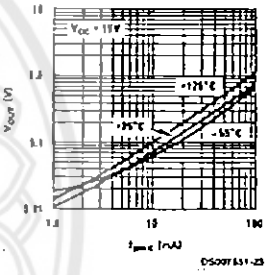
Low Output Voltage vs Output Sink Current



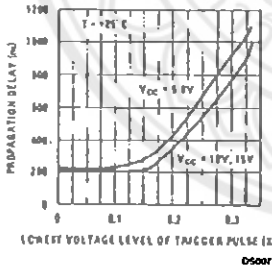
Low Output Voltage vs Output Sink Current



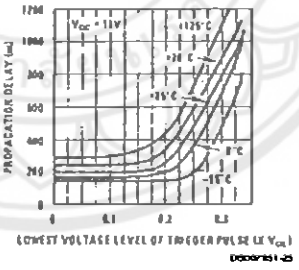
Low Output Voltage vs Output Sink Current



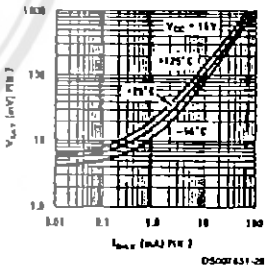
Output Propagation Delay vs Voltage Level of Trigger Pulse



Output Propagation Delay vs Voltage Level of Trigger Pulse

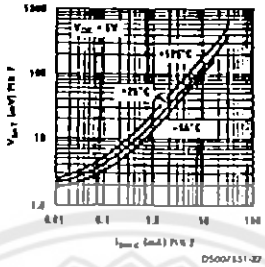


Discharge Transistor (Pin 7) Voltage vs Sink Current



Typical Performance Characteristics (Continued)

Discharge Transistor (Pin 7) Voltage vs Sink Current



Applications Information

MONOSTABLE OPERATION

In this mode of operation, the timer functions as a one-shot (*Figure 1*). The external capacitor is initially held discharged by a transistor inside the timer. Upon application of a negative trigger pulse of less than 1/3 V_{CC} to pin 2, the flip-flop is set which both releases the short circuit across the capacitor and drives the output high.

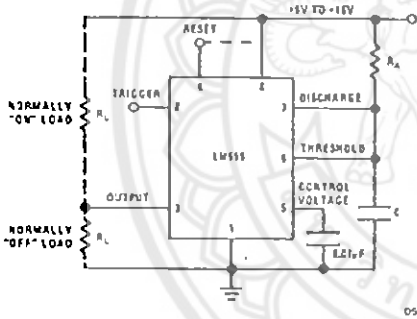
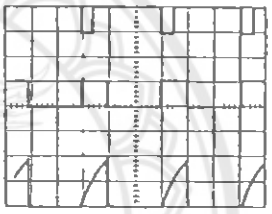


FIGURE 1. Monostable

The voltage across the capacitor then increases exponentially for a period of $t = 1.1\ R_A\ C$, at the end of which time the voltage equals 2/3 V_{CC}. The comparator then resets the flip-flop which in turn discharges the capacitor and drives the output to its low state. *Figure 2* shows the waveforms generated in this mode of operation. Since the charge and the threshold level of the comparator are both directly proportional to supply voltage, the timing internal is independent of supply.



V_{CC} = 5V
TIME = 0.1 ms/Div.
R_A = 9.1 kΩ
C = 0.01 µF

Top Trace: Input 5V/Div.
Middle Trace: Output 5V/Div.
Bottom Trace: Capacitor Voltage 2V/Div.

FIGURE 2. Monostable Waveforms

During the timing cycle when the output is high, the further application of a trigger pulse will not effect the circuit so long as the trigger input is returned high at least 10 µs before the end of the timing interval. However the circuit can be reset during this time by the application of a negative pulse to the reset terminal (pin 4). The output will then remain in the low state until a trigger pulse is again applied.

When the reset function is not in use, it is recommended that it be connected to V_{CC} to avoid any possibility of false triggering.

Figure 3 is a nomograph for easy determination of R, C values for various time delays.

NOTE: In monostable operation, the trigger should be driven high before the end of timing cycle.

Applications Information (Continued)

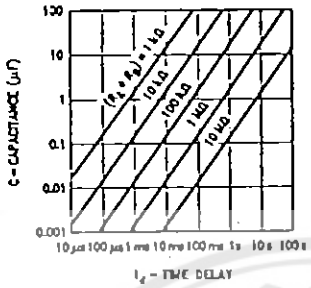


FIGURE 3. Time Delay

ASTABLE OPERATION

If the circuit is connected as shown in *Figure 4* (pins 2 and 6 connected) it will trigger itself and free run as a multivibrator. The external capacitor charges through $R_A + R_B$ and discharges through R_B. Thus the duty cycle may be precisely set by the ratio of these two resistors.

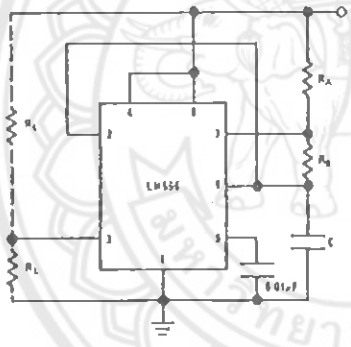
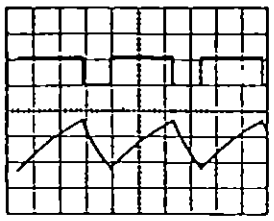


FIGURE 4. Astable

In this mode of operation, the capacitor charges and discharges between 1/3 V_{CC} and 2/3 V_{CC}. As in the triggered mode, the charge and discharge times, and therefore the frequency are independent of the supply voltage.

Figure 5 shows the waveforms generated in this mode of operation.



V_{CC} = 5V
TIME = 20 μs/DIV.
R_A = 3.9 kΩ
R_B = 3 kΩ
C = 0.01 μF

Top Trace: Output 5V/Div.
Bottom Trace: Capacitor Voltage 1V/Div.

FIGURE 5. Astable Waveforms

The charge time (output high) is given by:

$$t_1 = 0.693\ (R_A + R_B)\ C$$

And the discharge time (output low) by:

$$t_2 = 0.693\ (R_B)\ C$$

Thus the total period is:

$$T = t_1 + t_2 = 0.693\ (R_A + 2R_B)\ C$$

The frequency of oscillation is:

$$f = \frac{1}{T} = \frac{1.44}{(R_A + 2R_B)\ C}$$

Figure 6 may be used for quick determination of these RC values.

The duty cycle is:

$$D = \frac{R_B}{R_A + 2R_B}$$

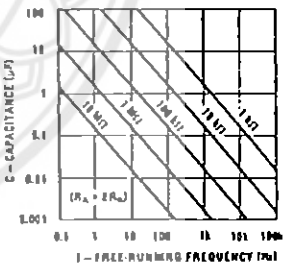
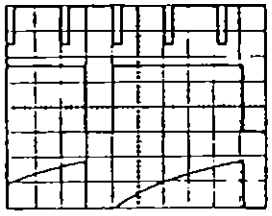


FIGURE 6. Free Running Frequency

FREQUENCY DIVIDER

The monostable circuit of *Figure 1* can be used as a frequency divider by adjusting the length of the timing cycle. *Figure 7* shows the waveforms generated in a divide by three circuit.

Applications Information (Continued)



V_{CC} = 5V
TIME = 20 µs/DIV.
R_A = 9.1 kΩ
C = 0.01 µF

Top Trace: Input 4V/Div.
Middle Trace: Output 2V/Div.
Bottom Trace: Capacitor 2V/Div.

FIGURE 7. Frequency Divider

PULSE WIDTH MODULATOR

When the timer is connected in the monostable mode and triggered with a continuous pulse train, the output pulse width can be modulated by a signal applied to pin 5. *Figure 8* shows the circuit, and in *Figure 9* are some waveform examples.

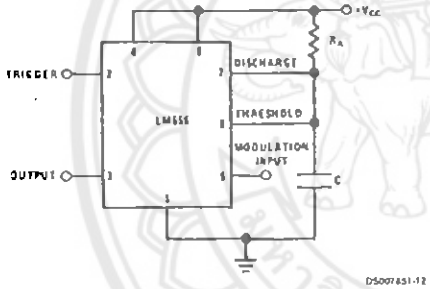
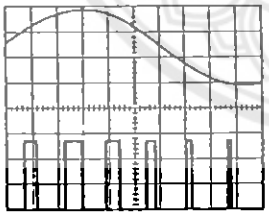


FIGURE 8. Pulse Width Modulator



V_{CC} = 5V
TIME = 0.2 ms/DIV.
R_A = 9.1 kΩ
C = 0.01 µF

Top Trace: Modulation 1V/Div.
Bottom Trace: Output Voltage 2V/Div.

FIGURE 9. Pulse Width Modulator

PULSE POSITION MODULATOR

This application uses the timer connected for astable operation, as in *Figure 10*, with a modulating signal again applied to the control voltage terminal. The pulse position varies with the modulating signal, since the threshold voltage and hence the time delay is varied. *Figure 11* shows the waveforms generated for a triangle wave modulation signal.

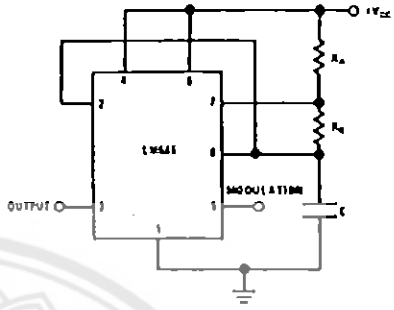
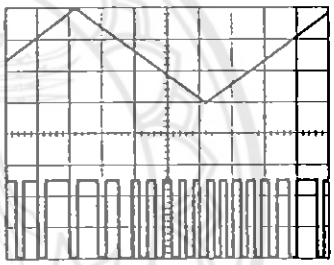


FIGURE 10. Pulse Position Modulator



V_{CC} = 5V
TIME = 0.1 ms/DIV.
R_A = 3.9 kΩ
R_B = 3 kΩ
C = 0.01 µF

Top Trace: Modulation Input 1V/Div.
Bottom Trace: Output 2V/Div.

FIGURE 11. Pulse Position Modulator

LINEAR RAMP

When the pullup resistor, R_A, in the monostable circuit is replaced by a constant current source, a linear ramp is generated. *Figure 12* shows a circuit configuration that will perform this function.

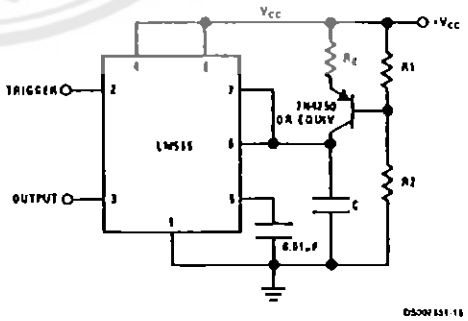


FIGURE 12.

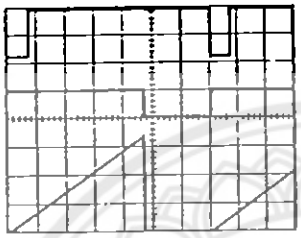
Applications Information (Continued)

Figure 13 shows waveforms generated by the linear ramp. The time interval is given by:

$$T = \frac{2/3\ V_{CC}\ R_E\ (R_1 + R_2)\ C}{R_1\ V_{CC} - V_{BE}\ (R_1 + R_2)}$$

$$V_{BE} \approx 0.6V$$

$$V_{BE} \equiv 0.6V$$



V_{CC} = 5V
TIME = 20 µs/DIV.
R_1 = 47 kΩ
R_2 = 100 kΩ
R_E = 2.7 kΩ
C = 0.01 µF

Top Trace: Input 3V/Div.
Middle Trace: Output 5V/Div.
Bottom Trace: Capacitor Voltage 1V/Div.

FIGURE 13. Linear Ramp

50% DUTY CYCLE OSCILLATOR

For a 50% duty cycle, the resistors R_A and R_B may be connected as in *Figure 14*. The time period for the output high is the same as previous, $t_1 = 0.693\ R_A\ C$. For the output low it is t_2 =

$$\left\{(R_A R_B)/(R_A + R_B)\right\}\ C\ ln\left\{\frac{R_B - 2R_A}{2R_B - R_A}\right\}$$

Thus the frequency of oscillation is

$$f = \frac{1}{t_1 + t_2}$$

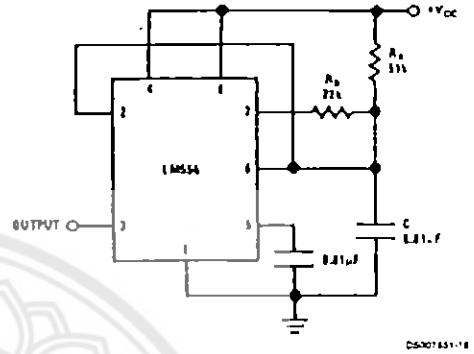


FIGURE 14. 50% Duty Cycle Oscillator

Note that this circuit will not oscillate if R_B is greater than 1/2 R_A because the junction of R_A and R_B cannot bring pin 2 down to 1/3 V_{CC} and trigger the lower comparator.

ADDITIONAL INFORMATION

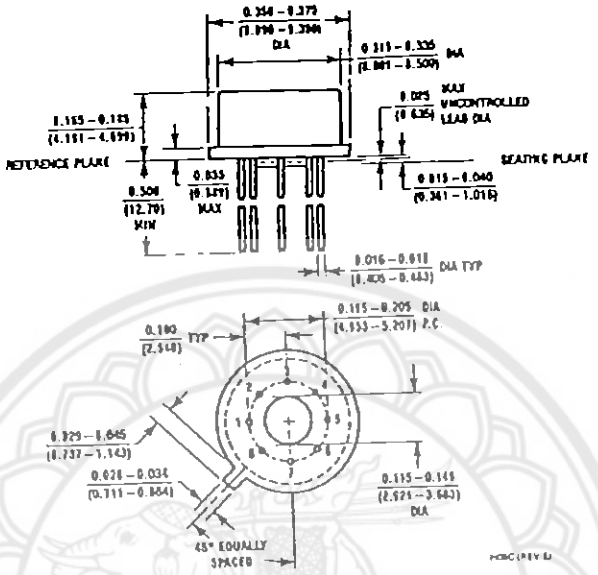
Adequate power supply bypassing is necessary to protect associated circuitry. Minimum recommended is 0.1 µF in parallel with 1 µF electrolytic.

Lower comparator storage time can be as long as 10 µs when pin 2 is driven fully to ground for triggering. This limits the monostable pulse width to 10 µs minimum.

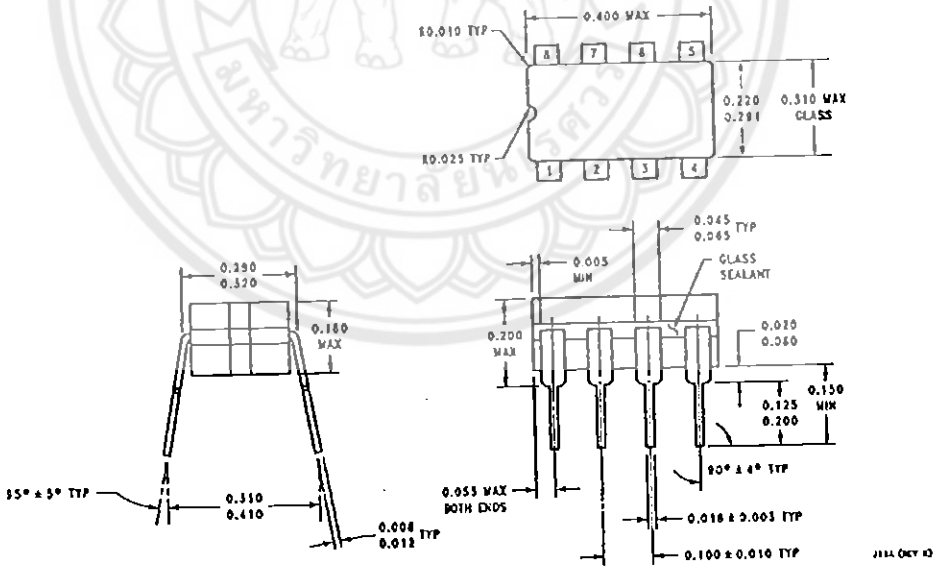
Delay time reset to output is 0.47 µs typical. Minimum reset pulse width must be 0.3 µs, typical.

Pin 7 current switches within 30 ns of the output (pin 3) voltage.

Physical Dimensions inches (millimeters)

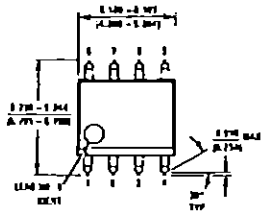


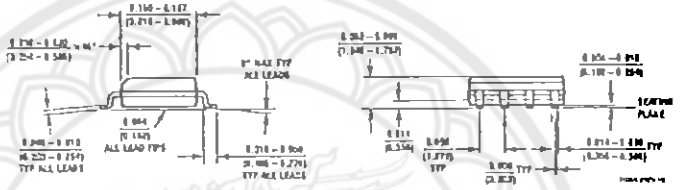
Metal Can Package (H)
Order Number LM555H or LM555CH
NS Package Number H08C



Ceramic Dual-In-Line Package (J)
Order Number LM555J or LM555CJ
NS Package Number J08A

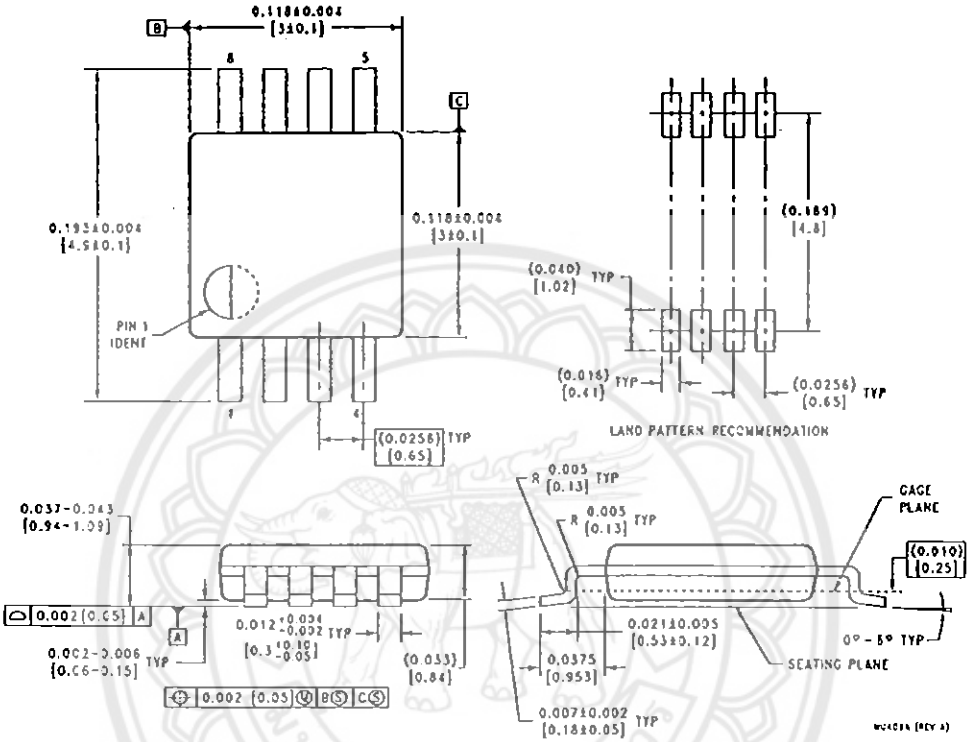
Physical Dimensions inches (millimeters) (Continued)





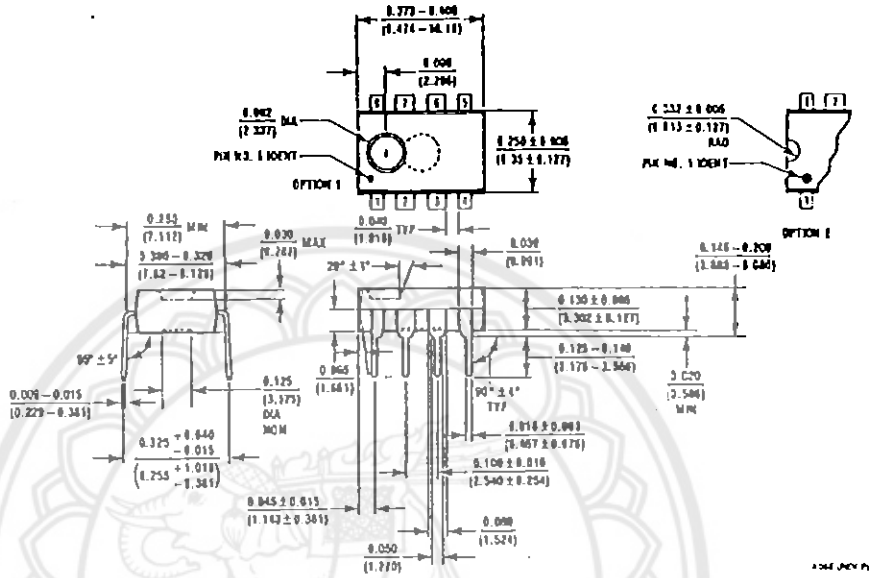
Small Outline Package (M)
Order Number LM555CM
NS Package Number M08A

Physical Dimensions inches (millimeters) (Continued)



8-Lead (0.118" Wide) Molded Mini Small Outline Package
Order Number LM555CMM
NS Package Number MUA08A

Physical Dimensions inches (millimeters) (Continued)



Molded Dual-In-Line Package (N)
Order Number LM555CN
NS Package Number N08E

LIFE SUPPORT POLICY

NATIONAL'S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS WRITTEN APPROVAL OF THE PRESIDENT OF NATIONAL SEMICONDUCTOR CORPORATION. As used herein:

1. Life support devices or systems are devices or systems which, (a) are intended for surgical implant into the body, or (b) support or sustain life, and whose failure to perform when properly used in accordance with instructions for use provided in the labeling, can be reasonably expected to result in a significant injury to the user.
2. A critical component in any component of a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness.

National Semiconductor Corporation
1111 West Bardin Road
Arlington, TX 76017
Tel: 1(800) 272-9959
Fax: 1(800) 737-7018

National Semiconductor Europe
Fax: (+49) 0-180-530 85 86
Email: onjwge@tevm2.nsc.com
Deutsch Tel: (+49) 0-180-530 85 85
English Tel: (+49) 0-180-532 78 32
Français Tel: (+49) 0-180-532 93 58
Italiano Tel: (+49) 0-180-534 16 80

National Semiconductor Hong Kong Ltd.
13th Floor, Straight Block,
Ocean Centre, 5 Canton Rd.
Tsimshatsui, Kowloon
Hong Kong
Tel: (852) 2737-1600
Fax: (852) 2736-9960

National Semiconductor Japan Ltd.
Tel: 81-043-299-2308
Fax: 81-043-299-2408

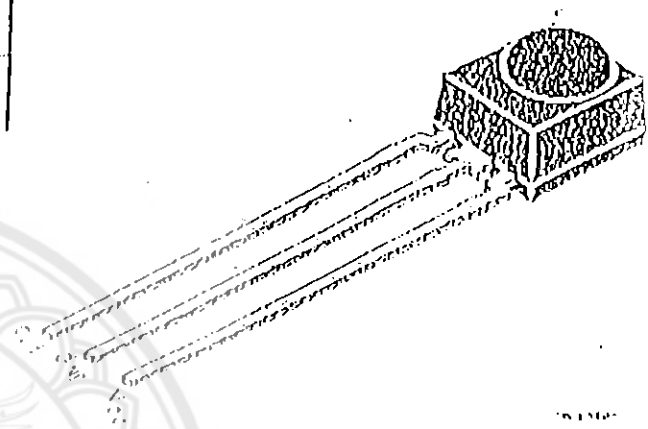
www.national.com

National does not assume any responsibility for use of any circuitry described, no circuit patent licenses are implied and National reserves the right at any time without notice to change said circuitry and specifications.

Photo Modules for PCM Remote Control Systems

Available types for different carrier frequencies

Type	f_0	Type	f_0
TSOP1830	30 kHz	TSOP1833	33 kHz
TSOP1836	36 kHz	TSOP1837	36.7 kHz
TSOP1838	38 kHz	TSOP1840	40 kHz
TSOP1856	56 kHz		



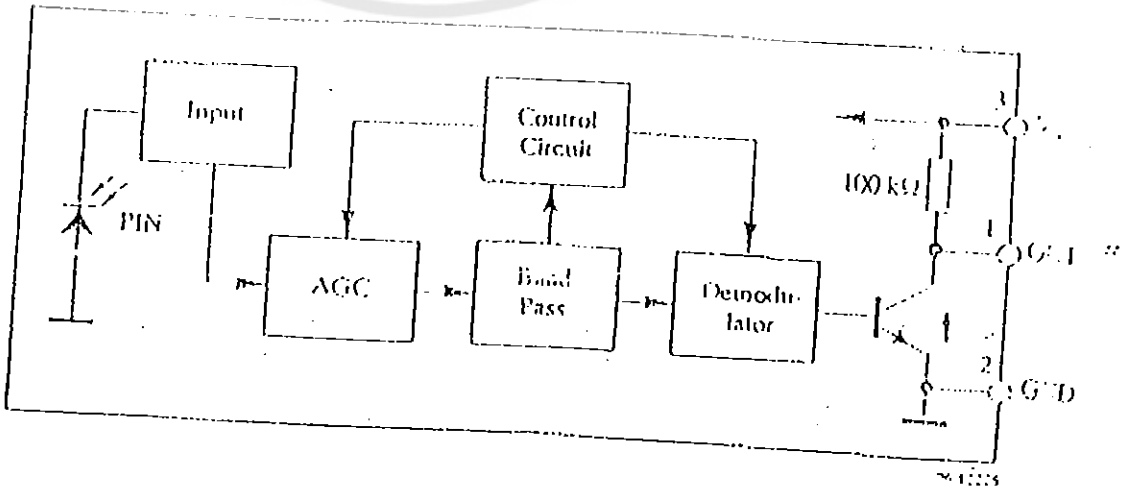
Description

The TSOP18.. - series are miniaturized receivers for infrared remote control systems. PIN diode and preamplifier are assembled on lead frame, the epoxy package is designed as IR filter.
The demodulated output signal can directly be decoded by a microprocessor. The main benefit is the reliable function even in disturbed ambient and the protection against uncontrolled output pulses.

Features

- Photo detector and preamplifier in one package
- Output active low
- Internal filter for PCM frequency
- Advanced immunity against ambient light
- Improved shielding against electric field disturbance
- 3 - 6 Volt supply voltage, low power consumption
- TTL and CMOS compatibility
- Intelligent AGC

Block Diagram



Absolute Maximum Ratings

T_{amb} = 25 °C

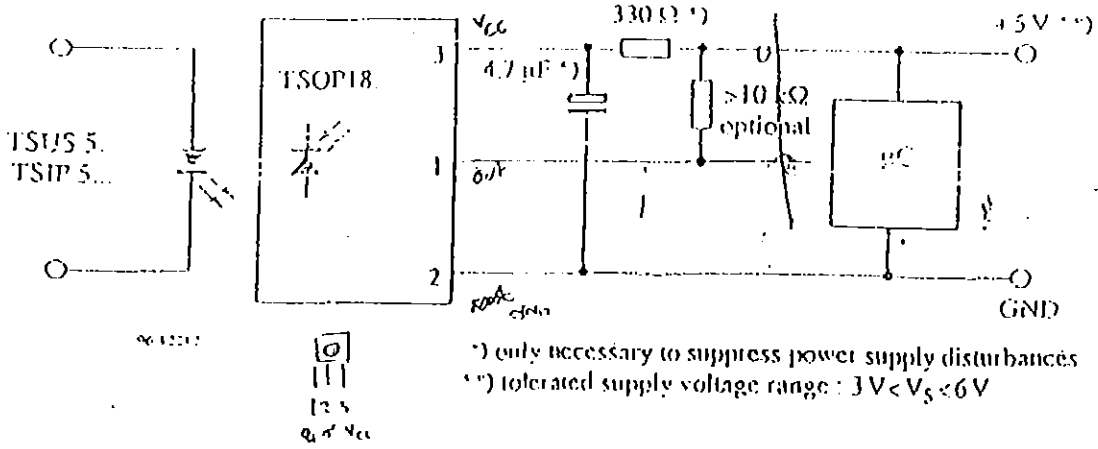
Parameter	Test Conditions	Symbol	Value	Unit
Supply Voltage	(Pin 3)	V_S	-0.3...6.0	V
Supply Current	(Pin 3)	I_S	5	mA
Output Voltage	(Pin 1)	V_O	-0.3...6.0	V
Output Current	(Pin 1)	I_O	5	mA
Junction Temperature		T_j	100	°C
Storage Temperature Range		T_{stg}	-25...+85	°C
Operating Temperature Range		T_{amb}	-25...+85	°C
Power Consumption	(T_{amb} ≤ 85 °C)	P_{tot}	50	mW
Soldering Temperature	t ≤ 10 s, 1 mm from case	T_{sd}	260	°C

Basic Characteristics

T_{amb} = 25 °C

Parameter	Test Conditions	Symbol	Min	Typ	Max	Unit
Supply Current (Pin 3)	V_S = 5 V, E_v = 0	I_{SD}	0.9	1.2	1.8	mA
	V_S = 5 V, E_v = 40 klx, sunlight	I_{SH}		1.6		mA
Transmission Distance	E_v = 0, test signal see fig.3, IR diode TSIP5201, I_F = 300 mA	d		30		m
Output Voltage Low (Pin 1)	I_{OSL} = 0.5 mA, E_e = 0.7 mW/m², (= I_O	V_{OSL}			250	mV
Irradiance (30 - 40 kHz)	Pulse width tolerance: $t_{pi} - 5/f_o < t_{po} < t_{pi} + 5/f_o$, test signal (see fig.3)	$E_{e\,min}$		0.4	0.5	mW/m²
Irradiance (56 kHz)	Pulse width tolerance: $t_{pi} - 5/f_o < t_{po} < t_{pi} + 5/f_o$, test signal (see fig.3)	$E_{e\,min}$		0.5	0.7	mW/m²
Irradiance		$E_{e\,max}$	30			W/m²
Directivity	Angle of half transmission distance	$\varphi_{1/2}$		±45		deg

Application Circuit



*) only necessary to suppress power supply disturbances
**) tolerated supply voltage range : 3 V < V_S < 6 V

Typical Characteristics (T_{amb} = 25°C unless otherwise specified)

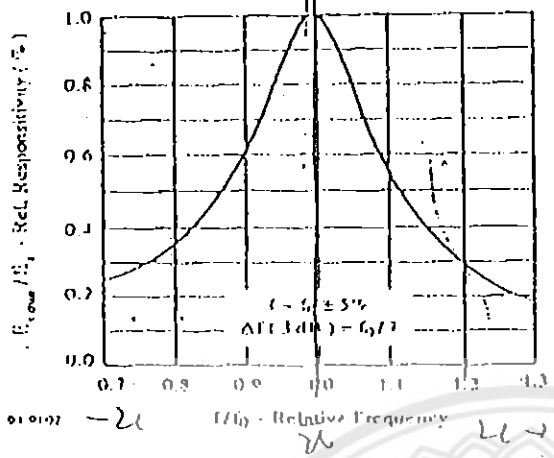


Figure 1. Frequency Dependence of Responsivity

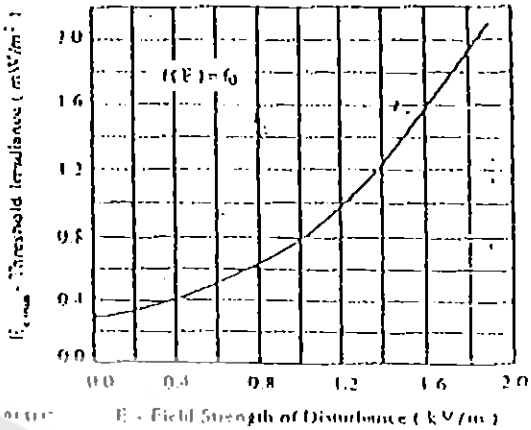


Figure 4. Sensitivity vs. Electric Field Disturbances

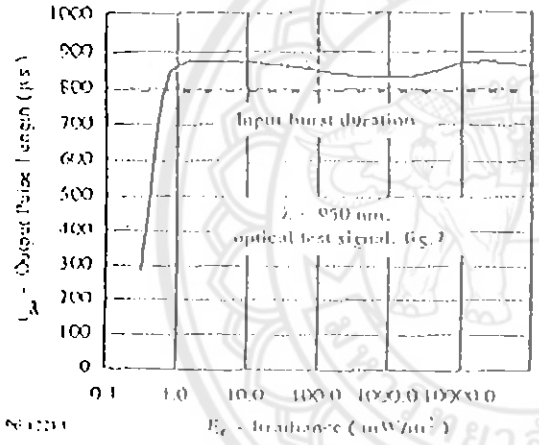


Figure 2. Sensitivity in Dark Ambient

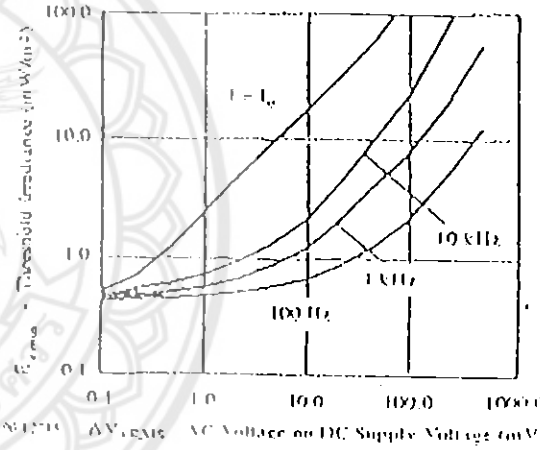


Figure 5. Sensitivity vs. Supply Voltage Disturbances

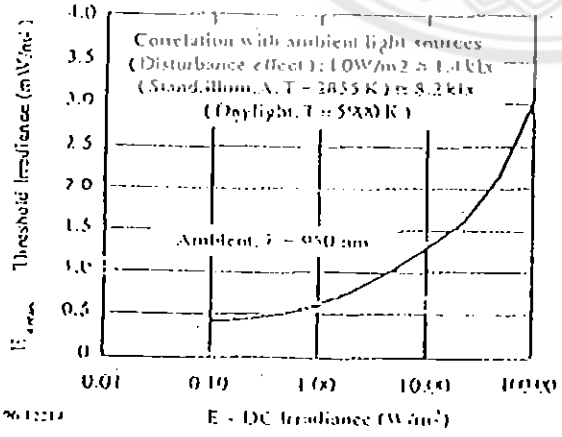


Figure 3. Sensitivity in Bright Ambient

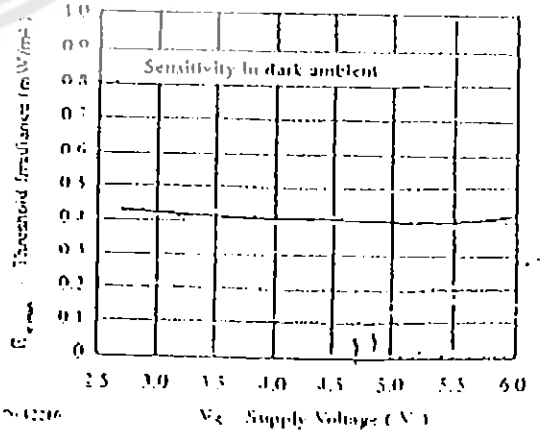


Figure 6. Sensitivity vs. Supply Voltage

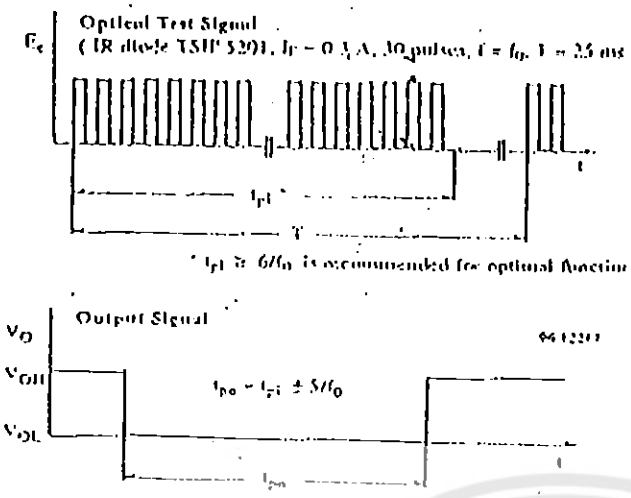


Figure 7 Output Function

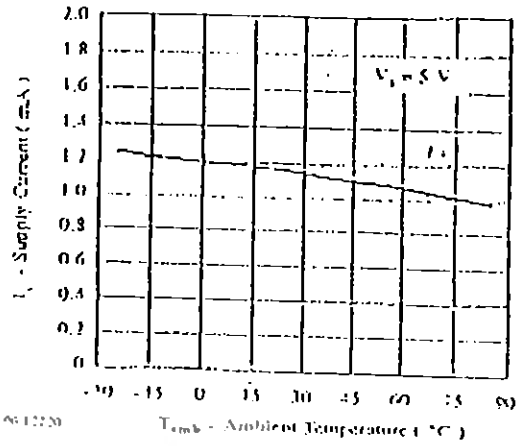


Figure 10 Supply Current vs. Ambient Temperature

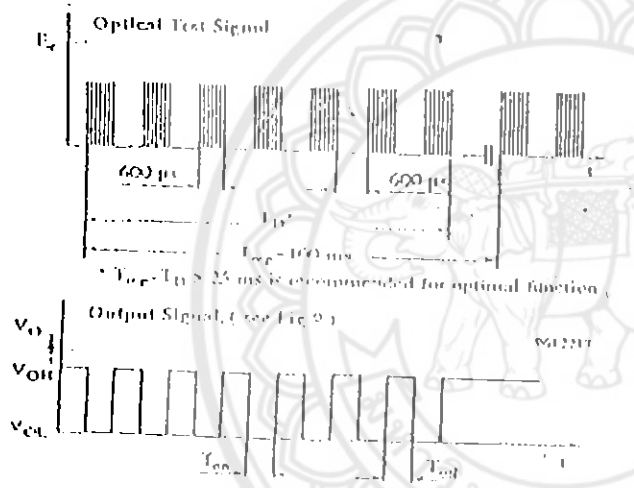


Figure 8 Output Function

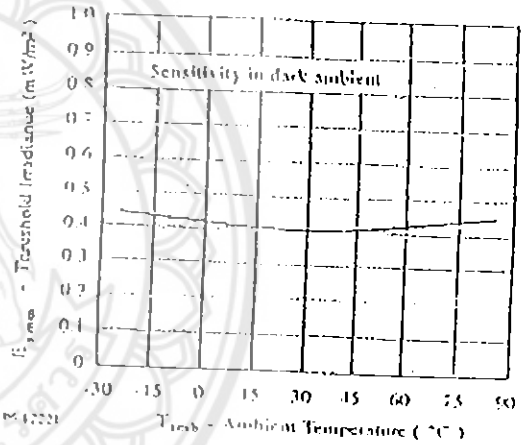


Figure 11 Sensitivity vs. Ambient Temperature

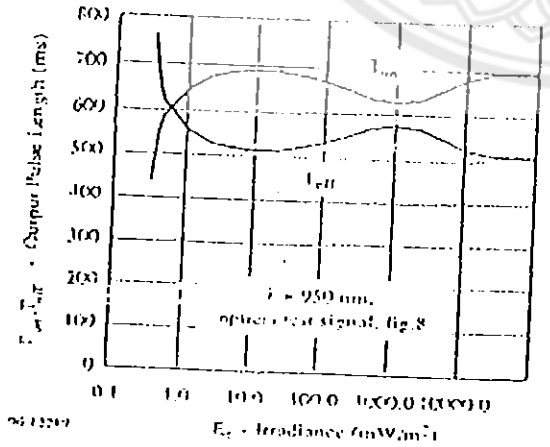


Figure 9 Output Pulse Diagram

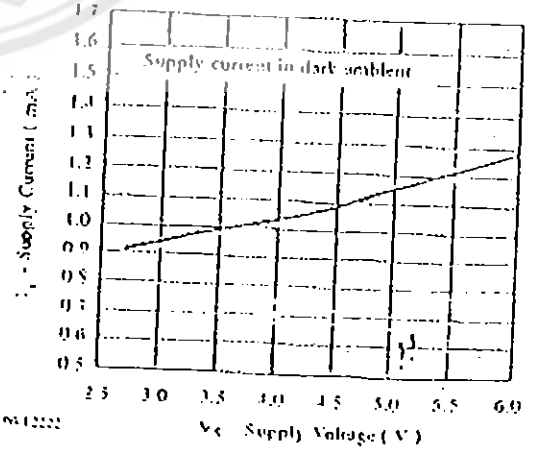


Figure 12. Supply Current vs. Supply Voltage

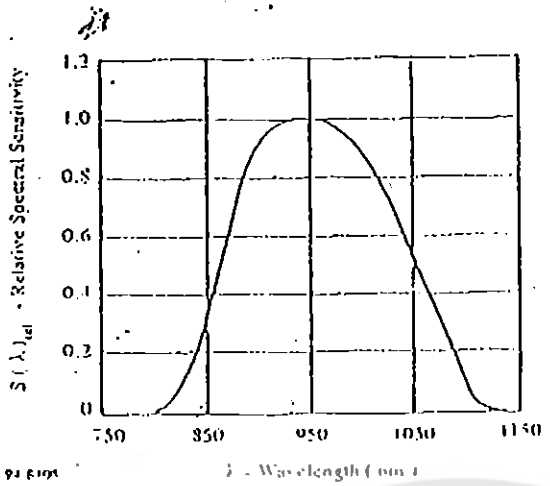


Figure 13. Relative Spectral Sensitivity vs Wavelength

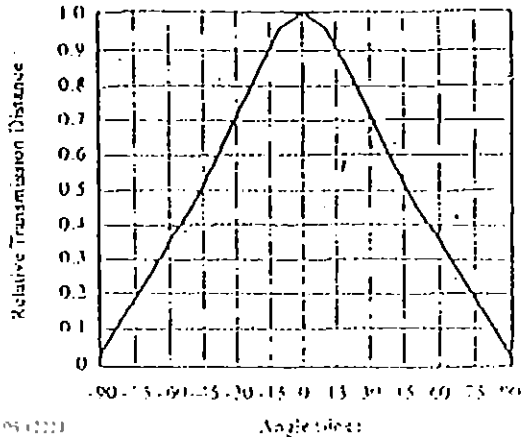
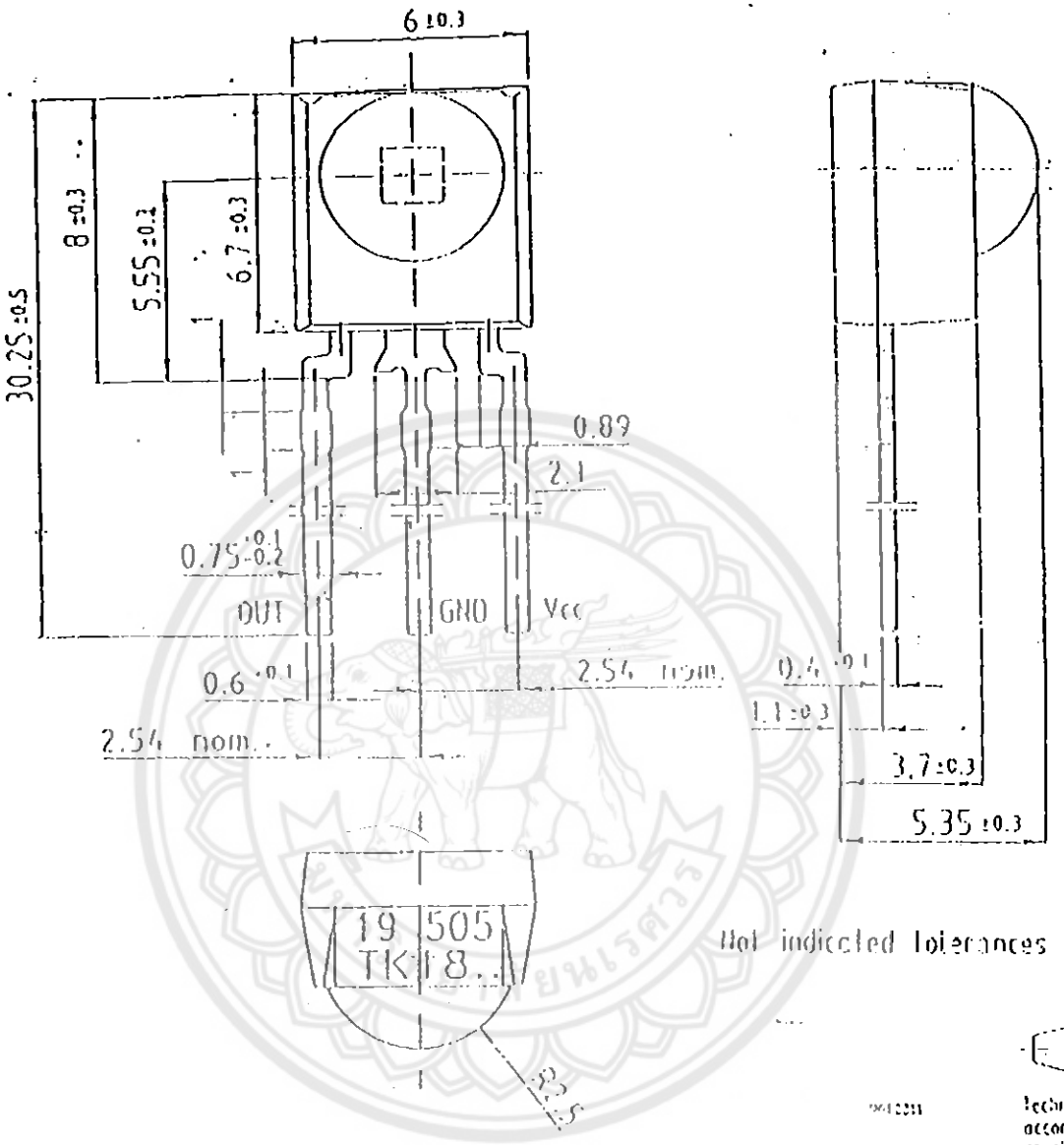


Figure 14. Directivity

TSOP18..

Dimensions in mm



Not indicated tolerances ±0.7

Technical drawings according to DIN specifications